Summer 2017

ItsBlue: A Distributed Bluetooth-Based Framework for Intelligent Transportation Systems

Ahmed Awad Alghamdi
Old Dominion University, aalghamd@cs.odu.edu

ITSBLUE

A DISTRIBUTED BLUETOOTH-BASED FRAMEWORK FOR INTELLIGENT TRANSPORTATION SYSTEMS

by

AHMED AWAD ALGHAMDI
B.S. June 2005, Al Imam Mohammad Ibn Saud Islamic University, Saudi Arabia
M.S. June 2009, The University of Melbourne, Australia

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2017

Approved by:

_____
Dr. Tamer Nadeem (Director)

_____
Dr. Kurt Maly (Member)

_____
Dr. Ravi Mukkamala (Member)

_____
Dr. Mecit Cetin (Member)

ABSTRACT

ITSBLUE
A DISTRIBUTED BLUETOOTH-BASED FRAMEWORK FOR INTELLIGENT TRANSPORTATION SYSTEMS

AHMED AWAD ALGHAMDI
Old Dominion University, 2017
Director: Dr. Tamer Nadeem

Inefficiency in transportation networks is having an expanding impact, at a variety of levels. Transportation authorities expect increases in delay hours and in fuel consumption and, consequently, the total cost of congestion. Nowadays, Intelligent Transportation Systems (ITS) have become a necessity in order to alleviate the expensive consequences of the rapid demand on transportation networks. Since the middle of last century, ITS have played a significant role in road safety and comfort enhancements. However, the majority of state of the art ITS are suffering from several drawbacks, among them high deployment costs and complexity of maintenance. Over the last decade, wireless technologies have reached a wide range of daily users. Today's Mobile devices and vehicles are now heavily equipped with wireless communication technologies. Bluetooth is one of the most widely spread wireless technologies in current use. Bluetooth technology has been well studied and is broadly employed to address a variety of challenges due to its cost-effectiveness, data richness, and privacy perverseness, yet Bluetooth utilization in ITS is limited to certain applications. However, Bluetooth technology has a potential far beyond today's ITS applications. In this dissertation, we introduce itsBlue, a novel Bluetooth-based framework that can be used to provide ITS researchers and engineers with desired information. In the itsBlue framework, we utilize Bluetooth technology advantages to collect road user data from unmodified Bluetooth devices, and we extract a variety of traffic statistics and information to satisfy ITS application requirements in an efficient and cost-effective way.

The itsBlue framework consists of data collection units and a central computing unit. The itsBlue data collection unit features a compact design that allows for stationary or mobile deployment in order to extend the data collection area. Central computing units aggregate obtained road user data and extract a number of Bluetooth spatial and temporal features. Road users' Bluetooth features are utilized in a novel way

to determine traffic-related information, such as road user context, appearance time, vehicle location and direction, etc. Extracted information is provided to ITS applications to generate the desired transportation services. Applying such a passive approach involves addressing several challenges, like discovering on-board devices, filtering out data received from vehicles out of the target location, or revealing vehicle status and direction.

Traffic information provided by the itsBlue framework opens a wide to the development of a wide range of ITS applications. Hence, on top of the itsBlue framework, we develop a pack of intersection management applications that includes pedestrians' volume and waiting times, as well as vehicle queue lengths and waiting times. Also, we develop a vehicle trajectory reconstruction application.

The itsBlue framework and applications are thoroughly evaluated by experiments and simulations. In order to evaluate our work, we develop an enhanced version of the UCBT Network Simulator 2 (NS-2). According to evaluation outcomes, itsBlue framework and applications evaluations show promising results. For instance, the evaluation results show that the itsBlue framework has the ability to reveal road user context with accuracy exceeding 95% in $25s$.

ACKNOWLEDGEMENTS

First and foremost, I praise God, the almighty, for His showers of blessings and bounties in my entire life, including his guidance of this research work.

I would like to express my sincere gratitude to my advisor, Dr. Tamer Nadeem, for the continuous support of my Ph.D. study and related research. His guidance has helped me to develop my researching and writing skills, in this dissertation.

My sincere thanks also go to Dr. Mecit Cetin for his helpful feedback and ideas, and for granting me access to his laboratory and his research facilities. I also would like to extend my thanks to the rest of my committee members, Dr. Kurt Maly and Dr. Ravi Mukkamala, for their insightful comments and encouragement. And I will never forget my late co-advisor Professor Hussien Abdelwahab who admitted me into the Ph.D. program and continuously supported me throughout my Ph.D. study until he passed away in December of 2016. God bless his soul.

During my Ph.D. study time at Old Dominion University, I have been surrounded by many colleagues and friends who have made it enjoyable. I would like to express my thanks to my colleagues in the Smart Wireless and Mobile Systems Research Lab (SWiMSys). And would like to extend my thanks to my friends in Norfolk, Virginia, who have eased my homesickness, especially to Abdullah Alhamli.

I would like to sincerely thank my uncle Ali Hajar, who inspires me. I am the oldest among my brothers and sisters, but he has been always there as an older brother for me. I also would like to extend my thanks to all of my uncles and aunts for their spiritual support, sincere wishes, and prayers.

I feel a deep sense of gratitude for my grandparents. I am named after one of my grandparents, and each one played a significant role in the development of my identity and in shaping the person who I am today. God bless their souls.

Special words of thanks go to my dear brothers Saad and Omar and to my dear sisters Areej and Hebah, for supporting me, believing in me, being proud of me, and taking care of my parents in my long times of absence.

This achievement would not have been possible without the support of my family. Words cannot express how grateful I am to my wife Areej Ramzi for the countless sacrifices that she has made to help me reach my goals and aspirations. My little man Areeb, thank you for our conversations about my work and the pride you have shown in me; they have been always strong motivations; thank you. My little

princess Jasmine, my face in your drawings , and your emotional comments about the drawings, have made this journey enjoyable; thank you. My sweet baby Mohammed, your cheerful face has been the cure after each long day of hard work; thank you.

No words of thanks can sum up the gratitude that I owe to my parents who raised me with unconditional love and support, and who encouraged me in all my pursuits. I owe them a great deal for the love, patience, and understanding they showed when I was away from home during my Masters and Ph.D. study time. I pay homage to them by dedicating my thesis to my mother, Maryam Hajar, and my father, the late Awad Alghamdi, under whose careful protection I have been able to reach this level of education and success. It was my father's wish to see me with a doctorate degree, and he has always been in my thoughts on this journey. He will remain there forever. God bless his soul.

I acknowledge Al Imam Mohammad Ibn Saud Islamic University, for providing a generous scholarship to assist me in the pursuit of my doctoral studies.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The demand for transportation is rapidly increasing and the numbers are indicating serious consequences. According to the United States Department of Transportation (US DOT) in the Transportation Statistics Annual Report 2016, congestion in the USA raised delay hours from 4.6 billion in 2000 to 6.9 billion in 2014 [1]. This jump in delay hours resulted in the consumption of 3.1 billion gallons of gas. Consequently, the total cost of congestion reached 160 billion US dollars. According to the 2015 Urban Mobility Scorecard that was issued by the Texas A&M Transportation Institute, the total cost of congestion is expected to grow to 192 billion US dollars in 2020 (Table 1). One of the most significant contributors to this increase is the anticipated increase in fuel consumption to 3.8 billion gallons, much of which will be wasted in 8.3 billion hours of delay [2].

The high demand on the transportation network is leading to serious complications at various levels including the environment, the economy and the public's and individuals' health. Environmentally, high fuel consumption used on an inefficient transportation network severely impacts air quality and contributes to global warming. *"The transportation sector is the second largest producer of greenhouse gas (GHG) emissions"*, notes the US DOT [1]. In 2014, about1300 million tons of CO_2 were emitted on USA roads [1]. The environmental impact was expected to expand due to an anticipated increase in fuel consumption in delays. Economically, the annual price of traffic congestion in the USA is expensive. Compared to 2014, the a 20% increase is expected in 2020's level of congestion. Besides the economic impact of road accidents in the USA, the number of highway vehicle fatalities grew from 34,641 in 2014 to 35,092 in 2015. Also, 2015 witnessed an increase in highway injuries – to 2.44 million. Furthermore, the unpleasant road experience is another adverse effect of the inefficient transportation network. The US DOT stated that, in 2014, travelers at the most congested areas had to allow 150% extra travel time during peak periods [1]. On the average, commuters spent 63 extra hours of travel time at congested areas, and 42 hours in any area, nationwide. The last number indicates over a 130% increase in time wasted on roads, compared to 1982 [2].

TABLE 1: Annual Congestion Delay and Costs in the USA.

Year	Delay per commuter (hours)	Total delay (billion hours)	Fuel wasted (billion gallons)	Total cost (billions U.S. dollars)
1982	18.0	1.80	0.50	42
1990	26.0	3.0	1.2	65
2000	37.0	5.20	2.10	114
2005	41.0	6.30	2.70	143
2006	42.0	6.40	2.80	149
2007	42.0	6.60	2.80	154
2008	42.0	6.60	2.40	152
2009	40.0	6.30	2.40	147
2010	40.0	6.40	2.50	149
2011	41.0	6.60	2.50	152
2012	41.0	6.70	3.0	154
2013	42.0	6.80	3.10	156
2014	42.0	6.90	3.10	160

Source: Texas A&M University, Texas Transportation Institute, 2015 Urban Mobility Scorecard [2]

1.1 STATE OF THE ART SOLUTIONS OVERVIEW

Enhancing transportation network efficiency has become both a necessity and a priority, in order to alleviate the aforementioned consequences. Transportation researchers and engineers have produced a vast literature regarding transportation enhancement and congestion control. Their effort led to emergence of the Intelligent Transportation Systems (ITS), in which a wide range of technologies are exploited to collect traffic data and to obtain the statistics required to address transportation challenges. ITS have played a significant role in boosting transportation network efficiency. For instance, Inductive Loop Detection (ILD) is a common ITS technology employed to enhance the performance of traffic lights and to monitor traffic fluidity. In addition, traffic statistics and information provided by ITS technologies will be fundamental pillars in future transportation plans and projects, including regulation revisions and infrastructure expansions. For example, GHG emissions data provided by ITS have drawn the attention of environmental authorities including the United States Environmental Protection Agency (US EPA), which has set several regulations to reduce GHG emissions, such as dedicating fast roadway lanes to buses and

carpools, in order to reduce the number of vehicles on the road.

Since the 1950s, dedicated ITS detection and monitoring technologies like magnetic field detectors or road surveillance cameras have significantly enhanced transportation network efficiency. For instance, magnetic field detection is a popular ITS technology that has been employed to detect vehicle passage across signalized intersections in order to dynamically control green light times. However, dedicated ITS detection technologies are suffering from number of disadvantages, such as inefficiency, expensive deployment and maintenance costs, and high privacy invasion. For instance, ILD's expensive and complicated installation and maintenance procedures limit its use to important locations only.

The evolution of mobile devices and wireless technologies encouraged ITS researchers and engineers to find new ways to collect traffic data. The ITS community employed electronic devices traveling on board vehicles to collect various kinds of data. Many of today's ITS technologies have adopted this approach. One example is the radio-frequency identification (RFID) transponder, which is used to identify RFID-hosting vehicles on tollway checkpoints. Later, the smartphones revolution led to emergence of participatory sensing, in which road users share their traffic data via third-party applications installed on their devices. The contribution of these technologies to ITS is undeniable. However, there are several drawbacks associated with them, such as data limitation and privacy invasion. For example, in participatory sensing, participants are assumed to have a special hardware or software, which narrows the circle of data sources.

In the following, we browse the major disadvantages of current ITS technologies:

- **Expensiveness**

  High price tags are associated to a wide range of most popular ITS technologies. For instance, aerial cameras are a commonly-used technology in road monitoring. However, to apply such system on a $500m$ roadway, seven video cameras are required. Each one cost a couple of thousand US dollars [3]. In addition to the sensor or detector tag price, employment of some ITS technologies include other expenses such as installation, maintenance, and site rent. For example, the ILD installation process involves pavement digging, wire burying, pull box connecting and road paving. This complicated installation process requires several costly tools and a number of well-paid professionals to complete the job. Further, installation or maintenance process of such ITS technologies requires

lane blocking for hours or days, which severely impacts traffic fluidity.

- **Inefficiency**

  The performance of various ITS is adversely affected by the surrounding environmental conditions and objects. Several ITS technologies such as infrared sensors and image-based sensors are impacted by inclement weather. Magnetic field detection technologies are prone to malfunctioning due to different reasons such as inclement weather, vehicle structure, and height. 30% out of 25000 ILDs in California are not working properly [4]. Also, evaluations of sound-based detection technologies like acoustic sensors show undercounting during peak and off-peak periods.

- **Data limitation**

  Available ITS technologies collect limited kinds of data. For example, magnetic field technologies like ILD detect vehicles; passage and count vehicles passing over. However, magnetic field detectors are unable to provide data about pedestrians, vehicle speeds, or trajectories. In addition, depending on unreliable source of data, like normal smartphones, users can receive incomplete or limited data. For example, smartphone users may stop the GPS service on their device to extend the battery life, which in turn will affect any ITS applications that collect GPS traces.

- **Privacy invasion**

  A wide range of current ITS technologies are invasive of privacy. For example, surveillance cameras are able to track road users and can record footage of their movements. Other technologies are able to go further. For instance, smartphone participatory sensing applications have access to a wide range of sensors such as GPS, microphones, cameras, inertial sensors, and dots which may cause a collection of high sensitivity data. Privacy concerns have encouraged several authorizes and governments to restrict the use of such technologies.

1.2 WIRELESS SENSING FEATURES

The undeniable drawbacks associated with many of state of the art ITS technologies have encouraged transportation researchers and engineers to find new ways to collect traffic data and to monitor the transportation network. In fact, the wide

spread of mobile wireless devices has motivated the transportation community to utilize wireless technologies to address a wide range of transportation challenges and to overcome legacy solution obstacles. Over the last decade, wireless technologies have become a hot research topic due to their capability to provide cost-effective solutions to challenges on various levels. In the following, we explain the main advantages of wireless technologies in addressing transportation challenges:

- **Ubiquity of mobile wireless devices**
  Nowadays, there are more than 4.7 billion cellphone devices in use worldwide, and over 2.3 billion of these devices are smartphones (FIG. 1). Additionally, over 12 million smartwatches were sold in 2016 [5]. Wireless communication is one of the essential services of mobile devices. The majority of today's mobile devices are provisioned with various wireless communication technologies like Wi-Fi, Bluetooth, and NFC. Furthermore, over the last decade, vehicles have become heavily equipped with driving assistance and entertainment systems. Such systems rely on wireless communication technologies to connect to user devices, to nearby vehicles or to roadside units. Additionally, autonomous driving vehicle development is showing rapid progress. According to the Autonomous Vehicle Disengagement Report issued by California DMV [6], Alphabet's (*i.e. Google's parent company*) autonomous driving vehicle, Waymo [7], drove over 635,000 miles in 2016. Wireless communication technologies are essential in this kind of vehicle, which allow it to collect an enormous amount of data from road users' wireless devices. The intensive existence of wireless technologies grants ITS applications access to a rich data source which, in turn, allows efficient solutions that can overcome current obstacles to be provided.

- **Cost-effectiveness**
  The low cost of wireless sensing devices allow the provision of cost-effective ITS applications. For instance, the Bluetooth-based ITS application cost is a fraction of the ILD system cost. Also, the installation or maintenance process is simpler and cheaper than the ILD system installation or maintenance process. Installation of a Bluetooth-based ITS application requires placing Bluetooth transceivers on appropriate road facilities (*e.g. traffic lights, road signs, and light poles*), and connecting them to a hosting computing unit. On the other hand, the ILD system installation or maintenance process involves roadwork

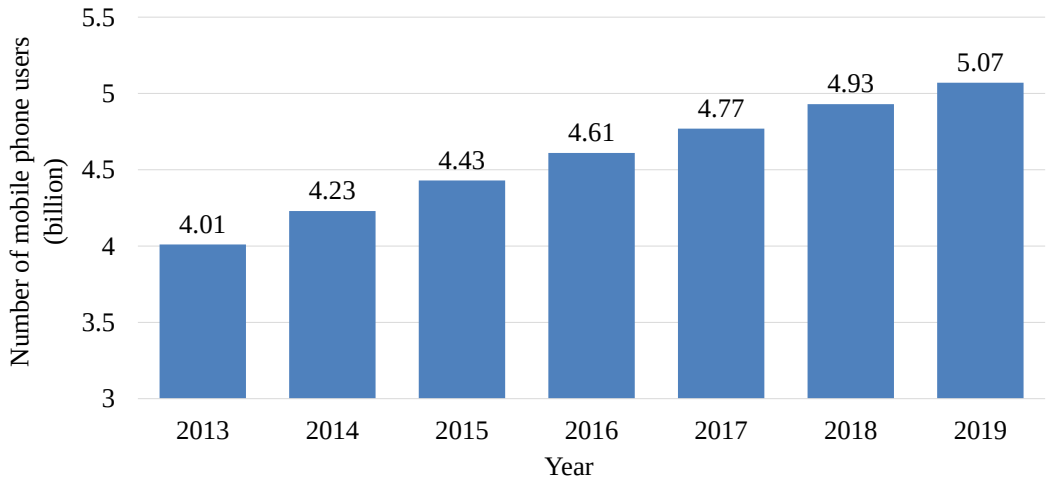


FIG. 1: Number of Cellphone Users Worldwide

Source: Statista The Statistics Portal [5]

and electricity connections. This complicated installation or maintenance process requires several costly tools and a number of well-paid professionals to do the work.

- **Privacy preservation**

  Wireless technologies ensure a high level of each user's privacy. Compared to legacy ITS technologies, wireless technologies are able to provide privacy preserving ITS solutions. For example, road monitoring cameras are able to track drivers and collect sensitive information such as plate numbers, and personal photos. Whereas, in wireless technologies, the only key piece of information that might be gathered is the transmitter networking identifier (*e.g. MAC address or Bluetooth address*). However, the transmitter networking ID is not considered to be a user's personal information. Besides, the majority of the state of the art wireless sensing applications scramble such information to ensure high privacy levels.

- **Data richness**

  Wireless technologies have been adopted to address a wide range of challenges due to their ability to extract various kinds of data. For example, wireless Radio Frequency (RF) technology has been widely adopted in indoor localization systems like RADAR [8] or Horus [9], in which wireless radio signals' spatial

characteristics are utilized to determine transmitter location. Further, wireless data collected by spatial and temporal sampling approaches are behind a wide range of today's technologies, such as travel time estimation, incident detection, and electronic toll collection.

1.3 BLUETOOTH FEATURES

One of the most widely used wireless technologies is Bluetooth. Bluetooth is the IEEE 802.15.1 wireless communication standard that is designed to be a low power consumer with a short range networking protocol, based on low-cost transceiver microchip, in order to provide a cable replacement technology. Since it was invented in the nineties, Bluetooth has become a hot research topic on various levels. In fact, Bluetooth technology features several advantages that make it an ideal choice for wide range of wireless sensing applications. For example, it makes data collection easier. Bluetooth communication standards facilitate collecting data from off-the-shelf Bluetooth-enabled devices with no third-party application or user involvement requirements. In the following, we briefly browse the main Bluetooth advantages:

- **High market penetration**
  Nowadays, 95% of adults in the USA have access to cellphones [10]. The majority of today's cellphones are Bluetooth-enabled. Actually, the number of various Bluetooth-enabled portable devices (*e.g. cellphones, tablets, smart watches, etc.*) is expected to grow to 10 billion devices in 2018 [11]. This shows to a 65% increase, compared to 2012. In addition, the Bluetooth connectivity feature is expected to reach 90% of new automobiles [12].

- **Data collection simplicity**
  Bluetooth communication protocols facilitate collecting Bluetooth data from devices in the vicinity. Bluetooth standards feature a discovery procedure, in which a Bluetooth device is able to search for nearby Bluetooth devices by broadcasting Bluetooth discovery messages, while Bluetooth visible devices in range are forced to reply with discovery response messages, which encompass their 48-bit Bluetooth addresses and other vital information. The Bluetooth discovery procedure eases the collection of data from off-the-shelf Bluetooth

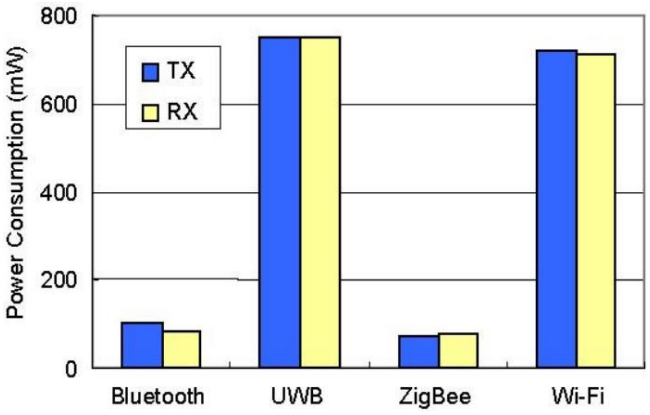


FIG. 2: Comparison of Several Wireless Technologies Power Consumption Rates

Source: A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi [15]

devices, with no third-party application. In addition, the Bluetooth discovery procedure allows Bluetooth devices that receive discovery messages to reply automatically without any user involvement. Furthermore, the discoverer Bluetooth device is able to collect an adequate number of discovery response messages over a short time with no connection established. According to [13], five Bluetooth devices are able to discover 20 nearby Bluetooth devices in $3s$. In Bluetooth version 4.0 and above, Bluetooth device discovery is completed in $3ms$ [14].

- **Low power consumption rate**

  Bluetooth's power consumption rate is lower than the majority of the wireless communication technologies, such as Wi-Fi or Ultra-WideBand (UWB). In a comparison between Bluetooth, ZigBee, Wi-Fi, and UWB, Bluetooth and ZigBee show very low power consumption rates, whereas Wi-Fi and UWB power consumption rates are six times higher (FIG. 2) [15]. However, the latest versions of Bluetooth (*i.e. Bluetooth version 4.0 and above*) have received massive power consumption enhancements. A recent study shows that Bluetooth Low Energy (BLE) achieves extremely low power consumption rate [16]. Comparing to ZigBee, the power consumption rate of BLE is 50% lower [17].

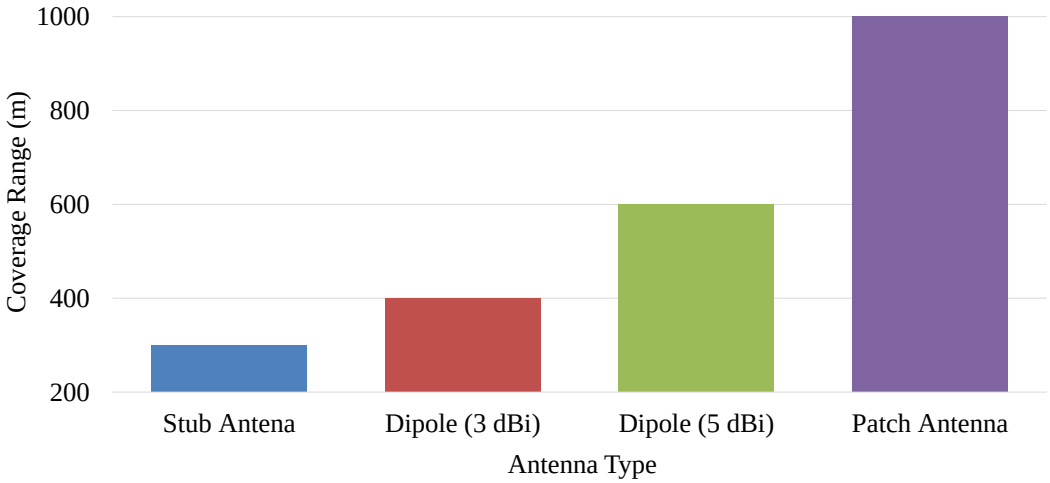


FIG. 3: Industrial Bluetooth Adapter Range with Multiple Antennas

- **Wide coverage range**
  Bluetooth's coverage range is able to be extended to wide ranges. In this research, we used a Class 1 industrial Bluetooth hardware from Sena Technologies, Inc. [18]. The Bluetooth adapter is shipped with a stub antenna which covers a range of $300m$. Further, the coverage range is able to be extended to $1000m$ using multiple power antennas (FIG. 3).

1.4 ITSBLUE

The distinctive advantages of wireless technologies, and precisely the Bluetooth, has encouraged researchers and engineers to employ it to address a wide range of challenges on various levels. However, Bluetooth utilization in the ITS domain is still limited, whereas, Bluetooth has a potential towards a wide range of transportation challenges. This, in turn, motivated us to utilize Bluetooth to build itsBlue, a cost-effective, low-maintenance and efficient Bluetooth-based framework to provide ITS researchers and application developers with real-time and historical traffic information. The itsBlue framework consists of data collection units distributed on a target area to collect required data, and a central computing unit that aggregates and manipulates the collected data to extract traffic-related information (FIG. 4). The itsBlue data collection units are compact computing units supplied with Bluetooth transceivers, and they are deployed on road facilities (*e.g. traffic lights, road signs, or light poles*) or are carried on vehicles to collect Bluetooth data from road users

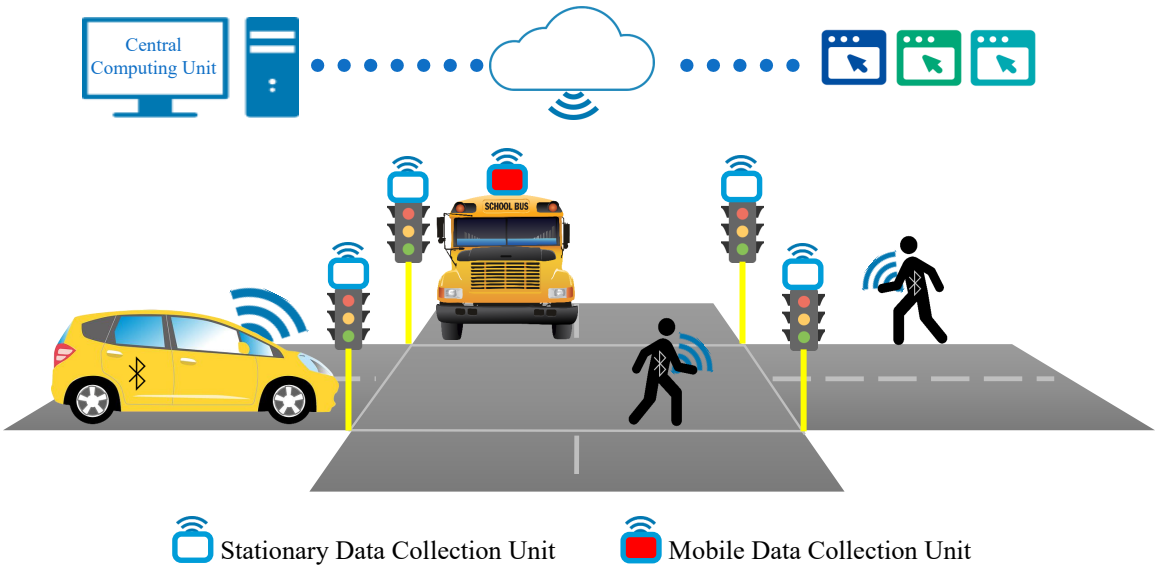


FIG. 4: itsBlue Framework Overview

(*i.e. drivers and pedestrians*). The itsBlue data collection unit collects Bluetooth data, filters it, synchronizes it with location data, and transfers it to the central computing unit. In addition, the data collection unit performs a part of the data manipulation to extract the location level information in a certain operation mode. The itsBlue central computing unit receives and aggregates the collected data, and then obtains the road user's basic Bluetooth features (*e.g. detected device appearance time, responsiveness, Bluetooth Radio Signal Strength (RSS) statistics, etc.*). Next, the central unit utilizes Bluetooth radio signal's spatial and temporal characteristics in a novel way to extract a road user's advanced features, like road user context (*i.e. vehicle rider or pedestrian*), vehicle locations at intersection, or moving vehicle directions. Road users' advanced features and other traffic data are provided to ITS applications. itsBlue facilitates traffic information delivery to ITS applications through an Application Programming Interface (API) in which a set of subroutine definitions are provided to allow ITS applications to consume the required information with isolation of networking complications.

In addition to the itsBlue framework, we developed several ITS applications on the top layer, a pack of intersection management applications and a passive vehicle trajectories reconstruction application. The intersection management applications pack includes the following services:

- Vehicle queue length.
- Vehicle waiting times.
- Pedestrians volume.
- Pedestrian waiting times.

The second application is a vehicle trajectories reconstruction application. This application employs moving vehicle locations and the directions provided by mobile data collection units to reconstruct detected vehicles trajectories. In this application, we show the novelty of utilizing the spatial and temporal characteristics of Bluetooth radio signals to extract the moving vehicle's street location and direction. In addition, we address several challenges to enhance reconstructed vehicle trajectories' correctness and completeness.

1.5 CONTRIBUTION

The following points summarize our contribution in this dissertation:

- The design and development of itsBlue, a passive Bluetooth-based framework to provide ITS applications with number of road users traffic information, in order to develop a wide-range transportation service in an efficient and cost-effective way.
- The utilization of Bluetooth technology potentials to produce an independent single site (*e.g. intersection*) ITS services. To the best of our knowledge, Bluetooth adoption in ITS is limited to a certain kind of applications, those in which a vehicle is sampled in two or more sites (*e.g. along a highway*) to extract specific statistics, such as travel time and speed.
- The novel utilization of Bluetooth radio signals' temporal and spatial characteristics to extract the following traffic-related features:
  - Road user context: Bluetooth radio signals received from a road user are utilized to extract a number of Bluetooth features (*e.g. RSS variance*) that allow itsBlue to differentiate between pedestrians and vehicle riders.
  - Vehicle location at signalized intersection: Wireless RF technology is applied for first time in ITS to locate vehicle spots at traffic light intersection.

  - Moving vehicle street location and direction: The features of Bluetooth radio signals received from a moving vehicle are utilized with an awareness of the receiving location in a novel way to determine the vehicle's street and direction.
  - Road user appearance time: Bluetooth messages received from a road user are utilized to determine the time elapsed in a certain location.

- The extension of the UCBT Bluetooth Network Simulator-2 (NS-2) to include a physical layer. Bluetooth simulators are rare and complicated to configure, due to the lack of documentation. UCBT is a NS-2 extension that enables Bluetooth simulations. However, it lacks a physical layer. Therefore, we implemented a physical layer into UCBT NS-2 that allows us to determine Bluetooth radio signals' vital measurements in order to evaluate itsBlue framework and services.

- The deployment of wireless-based ITS applications with no lane or road blocking. A variety of wireless sensing systems, especially RF-based ones, require a training data set which is collected during system deployment. Such a process involves road or lane blocking to obtain the required data from pre-defined devices placed on the road. To ensure traffic fluidity, we establish a novel training dataset collection approach that allows us to obtain the required data from moving vehicles.

1.6 DOCUMENT ORGANIZATION

The rest of the document is organized as follows. In the next chapter, we thoroughly discuss state of the art ITS technologies. Chapter 3 is devoted to presenting the itsBlue framework and its components. In Chapter 4, we describe data collection and processing. In Chapter 5, we explain basic Bluetooth features extraction. And in Chapter 6, the advanced road user extraction is illustrated. Chapter 7 describes the framework services provision to ITS application using itsBlue API. Chapter 8 is devoted to presenting ITS applications that have been developed using the itsBlue framework. In addition, chapter 8 includes the itsBlue framework and the ITS applications evaluation, and it presents our extended version of UCBT NS-2. Finally, in Chapter 9, we conclude the present work and discuss future directions.

CHAPTER 2

RELATED WORK

Over the past fifty years, the ITS revolution has produced much research and thousands of inventions in a variety of formats. In this chapter, we explore and discuss related state of the art work in ITS. Our proposed work relies on road users' detection to provide services. Thus, in this section, we browse the sensing technologies employed in the ITS domain. To enhance the clearness and comprehensiveness, related work is classified by technology. Every section introduces a sensing technology and discusses its advantages and disadvantages.

2.1 INDUCTIVE LOOPS DETECTION (ILD)

ILD is a widely adopted technology in ITS. Basically, ILD is an insulated wire buried in a shallow closed shape slot in the pavement, and connected to a pull box from a side where a lead-in wire connects the other side to the traffic control unit. The traffic control unit sends an electrical current through the loop, which generates a magnetic field. The traffic control unit continuously monitors the inductance level of the loop. When a metallic object (*i.e. vehicle*) stops or passes over the loop, the inductance level of the loop decreases, which triggers the traffic controller unit to announce a vehicle arrival [19]. Modern ILDs are able to classify vehicle profiles by utilizing the variance of the inductance level caused by detected objects [20, 21]. The high cost of installation and maintenance is the main advantage of ILD technology [19]. In addition, reported ILDs malfunction is about 30%. The associated error occurs due to several reasons, such as a vehicle's structure and height or inclement weather conditions [20].

2.2 MAGNETIC SENSORS

A magnetic sensor detects the change in the Earth's ambient magnetic field caused by the passage of a vehicle. The magnetic sensor consists of a cylinder containing coils which is placed under bridge or in a hole in the center of a road lane. The magnetic sensor detecting technique is very similar to that used by ILDs, even though

magnetic sensors outperforms ILDs in durability and installation and maintenance easiness. On the other hand, most of ILDs' functionality problems occur in magnetic sensors [19].

2.3 MICROWAVE SENSORS

Microwave sensors are usually deployed on top of traffic lights or on high poles. The microwave sensor transmits electromagnetic waves to detect vehicles in a similar way to ILDs. This kind of sensor is broadly adopted due to the wide range of applications that it serves. Compared to ILDs, the microwave sensor is lower in initiation cost and easier to maintain. The disadvantage of this sensor is interference with other microwave devices in vicinity [20].

2.4 ULTRASONIC SENSORS

The ultrasonic sensor transmits ultra sound waves at frequencies from 25KHz to 50KHz. Then it measures the reflected waves to detect vehicle passage. Ultrasonic sensors are usually installed on the ground, whereas they might be installed over the road for vehicle profiling purposes. Ultrasonic sensor maintenance costs are lower than the cost for ILDs and magnetic sensors. In addition, they show a high level of reliability and durability. However, ultrasonic sensor performance is adversely impacted by wind or high temperatures [19].

2.5 ACOUSTIC SENSORS

Acoustic sensors rely on a vehicle's noise to detect its passage. The acoustic sensor consists of array of microphones which listen continuously to audible sounds in the target area. Upon a vehicle's approach, its acoustic energy level is increased, which triggers the sensor to report the vehicle's arrival. For optimum performance, the microphones are deployed on the vehicle tires' level and their sensing sensitivity is set to frequencies between 50Hz and 2000Hz. However, acoustic sensors face performance issues in inclement weather and during peak and off-peak times [20, 19].

2.6 LASER SENSORS

Laser sensors illuminate a vehicle with a laser beam and analyze the reflected light to detect it and to discover its characteristics. Laser sensors are a widely adopted

technology due to their wide range of applications, such as speed detection, volume detection, and profile detection [20], even though laser sensing technology is not commonly used in several ITS applications such as vehicle queue length estimation.

2.7 INFRARED SENSORS

The infrared sensing technology concept is similar to laser sensing. An infrared sensor transmits an infrared ray into the target area and analyzes the energy reflected from a passing by vehicle to discover it. Infrared sensing technology is widely used, as it satisfies a wide range of application requirements. However, it is an expensive technology and its performance is adversely affected by inclement weather [21].

2.8 IMAGE-BASED SENSORS

Image-based sensing is one of the most broadly adopted techniques in pedestrian and vehicle detection applications. Image-based sensors consist of one or more video cameras that take a series of pictures for target area, and then transfer it to a processing unit that utilizes image processing techniques to discover pictured objects depending on pixel variance. Dollar et al. [22] investigated a number of image-based pedestrian detectors. The majority of the surveyed systems suffered from significant delays; the measured detection times were in tens of seconds per frame. However, there are a number of image-based pedestrian detection systems that have achieved higher speeds of 10-30 frames in few seconds, although these systems require costly computing infrastructures [23]. Further, the pedestrian detection True Positive Rate (TPR) of several state of the art image-based systems is around 50%. On the other hand, the majority of vehicle image-based detection systems rely on aerial or side images. Consequently, these systems are complicated and involve a number of classifiers and a long training process [24]. Ultimately, besides installation and calibration complications, image-based sensing techniques suffer from high computation costs and wide data transmission bandwidth requirements. Also, traffic congestion, vehicle speed, variation of light and inclement weather adversely impact most image-based vehicle detectors' performance [25].

2.9 PARTICIPATORY SENSING

Participatory sensing technology aims to utilize the ubiquity of smartphones to collect a user's context and the surrounding environment's data. Modern smartphones have become loaded with various sensors and communication technologies, which allow them to collect substantial data offering wide range of applications. Participatory sensing technology aims to utilize a user's collected data to extract ambient information (*e.g. air pollution level monitoring, ambient noise monitoring, etc.*), context, or personal information (*e.g. health monitoring, physical activities, social information, etc.*).

The diversity of sensors and wireless communication technologies available on smartphones nowadays lead to different sensing techniques. Using device inertial sensors (*e.g. accelerometer, gyroscope, or compass*) to detect user activities is one of the most common sensing practices in participatory sensing. Actually, several daily activities have distinguishable inertial sensors signatures (FIG. 5), which allows one to detect user context. Thus, participatory sensing ITS applications are able to infer journey information by utilizing data collected by an on-board device. For instance, Nericell [26] is an attempt to utilize a participatory sensing concept in ITS. The authors proposed a system to discover the street that the vehicle is moving on from the on-board device inertial sensor's signature. Then, the system finds the vehicle's location by using the phone Global System for Mobile Communications (GSM) signal characteristics. However, the proposed system evaluation showed $660m$ error 90% of the time.

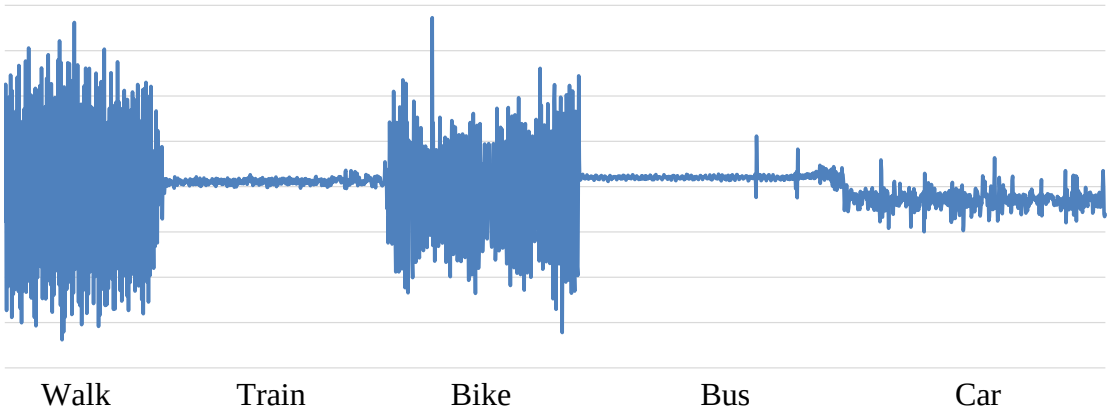


FIG. 5: Example of Inertial Sensors (Accelerometer) Activity Signatures.

Source: Detecting Vehicle Stops From Smartphone Accelerometer Data [27]

Moreover, the availability of wireless networking technologies on smartphones and such devices motivated researchers to employ them in participatory sensing applications. For example, Bluetooth has been utilized in participatory sensing applications to estimate the car congestion level on trains. The proposed system applies the Bayesian theorem to estimate the likelihood of transmitter existence on train cars, and then infers the congestion level from number devices located on that car [28]. In addition, QueueSense is a participatory sensing system that recognizes human waiting queues. It utilizes smart phone sensors and Bluetooth to continuously collect data in order to discover devices in queues. Then it uses the collected data to separate waiting lines using an SVM classifier, and finally it obtains the queue waiting times. [29]. Weppner et al. [30] investigated participatory sensing in crowd density level estimation in public locations. Their proposed system employed Bluetooth signals and GPS locations collected from volunteers' devices to extract a number of features in order to predict crowd level.

In fact, participatory sensing includes all of the practices of sensing by collecting data via third party hardware or software. Probe vehicle is one of the most common practices of participatory sensing in ITS domain. In definition, probe vehicle is an ITS technology designed for specific applications such as traffic monitoring, incident detection, route guidance, and queue length estimation [31, 32, 33]. Probe vehicle is a specific purpose hardware or software installed on a vehicle or an on-board device to collect real time trip data and then transfer it to an ITS application. Probe vehicle utilizes four techniques to collect data:

- **Signpost-Based Automatic Vehicle Location (AVL):** Probe vehicle communicates with transmitters deployed on signposts.
- **Automatic Vehicle Identification (AVI):** Vehicle is supplied with an on-board electronic tag that communicates with roadside transceivers to identify the vehicle and to obtain travel times between transceivers.
- **Ground-Based Radio Navigation:** It is similar to the Global Positioning System (GPS) concept, whereas radio towers collect data from probe vehicle in order to determine a location.
- **Cellular Geo-location:** Experimental technology aims to estimate travel time data by tracking cellular telephone call transmissions.

- **GPS:** On-board two-way GPS transceivers that continuously receive GPS satellite signals and transmit them to an ITS application that provides location-based services.

Probe vehicle is playing an essential role in various ITS applications, such as fleet management and transit agencies' applications [34].

Participatory sensing showed potential to assist in a variety of ITS challenges. However, participatory sensing suffers from several disadvantages, with a high level of privacy invasion being one of its major drawbacks. Participatory sensing applications have access to the majority of device resources (*e.g. inertial sensors, GPS, and cameras*) which violate users' privacy by accessing sensitive information such as GPS location. In addition, the high energy consumption rate associated with several sensors used in participatory sensing is another disadvantage. According to a recent study conducted on an Android platform system [35], in a certain time frame, GPS consumes about 15%, inertial sensors consume 10% and a camera drains about 20% of the battery power capacity. In addition, the success of participatory sensing applications relies on the number of volunteers who are willing to share their data, whereas a number of discouraging factors are associated with participatory sensing, such as privacy invasion, a high energy consumption rate, a high cellular data usage rate, or a third-party application prerequisite. These factors are enough to push a considerable number of users not to volunteer to participate in sensing applications. Regarding probe vehicle, the hardware on-board installation requirement makes it unsuitable choice for various ITS applications.

2.10 WIRELESS SENSING

Nowadays, wireless communication technologies have widely spread. Modern mobile devices include a variety of wireless communication means such as Wi-Fi, Bluetooth, ZigBee, and Near Field Communication (NFC). The availability of these well-defined standards has motivated researchers to employ wireless sensing to address several challenges. Wireless sensing applications utilize transmitted data packets and radio signal characteristics to detect transmitting devices and to determine several pieces of information about them and their surrounding areas. According to wireless communication standards, every data packet encompasses a transmitter identifier called a Media Access Control (MAC). This unique identifier allows wireless sensing applications to detect and track the transmitter. In addition, physical radio signal

characteristics include several features which are key to determining useful data. For example, radio signal transmission power that fades over distance allows wireless sensing applications to determine a transmitter's location.

In fact, the main advantage of wireless sensing over other sensing technologies that utilize wireless communication means (*e.g. participatory sensing*) is the ability to use unmodified user devices to collect data. Wireless sensing applications benefit from the wireless communication standards that govern wireless devices to collect data from off-the-shelf devices, with no modification or third-party application prerequisite. Wireless sensing applications utilize networking standard protocols to communicate with devices within range and to collect the required data. Actually, Wi-Fi and Bluetooth are the most popular technologies in wireless sensing applications, due to their high market penetrations. However, the Bluetooth neighbors' discovery process facilitates obtaining the required data from in-range devices, which grants Bluetooth an advantage over Wi-Fi.

Over the last decade, wireless sensing has employed Wi-Fi and Bluetooth to address various challenges. For instance, the absence of a GPS satellite signal indoors motivated researchers to investigate alternative technologies in order to provide location services. Wireless sensing technology has been one of the most suitable alternatives, due to its simplicity, cost-effectiveness, and data richness. Wireless sensing applications have adopted RF technology to estimate in-range devices' locations. The radio signal's received power is the key feature in determining transmitter location.

In the ITS domain, Bluetooth is extensively utilized to estimate travel times on highways. Simply, two or more Bluetooth transceivers are placed a bit apart on a highway to collect the Bluetooth radio signals from vehicles traveling by them. Then, the travel time is calculated as the difference between the signals' receiving times [36]. A number of researchers have studied the impact of Bluetooth transceiver placement and signal selection among received signals at a given sampling point. These researchers showed the impact of Bluetooth transceiver placement and recommended selecting the strongest and last-received Bluetooth signal to enhance the accuracy of travel time estimation [37, 38, 39]. Further, Bluetooth contributed to route estimation by collecting Bluetooth radio signals from vehicles passing by certain checkpoints, in order to obtain traffic information and estimate route travel time [40]. Moreover, a number of studies have addressed pedestrian assistance challenges using Bluetooth-based sensing. Universal Real-time Navigational Assistance (URNA) is

a Bluetooth-based navigation system for blind persons at signalized intersections. The system establishes a connection with a pedestrian Bluetooth device and sends it messages about intersection topology and traffic light signals [41].

To conclude, wireless sensing technologies are intensively utilized in localization applications to address the GPS high energy consumption rate and the absence of indoor signals, whereas wireless sensing technology utilization in ITS is limited to certain kinds of applications, such as travel time estimation on highways. However, wireless sensing technologies have the potential to address various ITS challenges.

2.11 CONCLUSION

ITS researchers and engineers have investigated a wide range of sensing technologies to address transportation challenges. However, the majority of the state of the art ITS technologies are suffering from different disadvantages. Magnetic field sensing and image-based sensing technologies are expensive to deploy and maintain. The performance of sound wave detection systems is adversely affected by surrounding environmental conditions, while inclement weather impacts the performance of infrared-based sensing applications. Participatory sensing is providing a competitive alternative by utilizing smartphones' sensors and wireless communication technologies. However, a number of disadvantages are associated with it, such as data limitations, high privacy invasion, and a third-party application requirement.

Wireless sensing, and Bluetooth precisely, has been well studied and broadly adopted to address a wide range of challenges in different domains. Yet Bluetooth is not well utilized in ITS. However, Bluetooth's features, like its low cost, data richness, and privacy preservative nature, make it an appropriate technology to address several ITS challenges. Therefore, in this research, we employ the Bluetooth wireless technology to introduce an efficient and cost-effective framework to provide the ITS community with required traffic statistics and information. Unlike other ITS applications, itsBlue is able to collect Bluetooth data from road users' commodity Bluetooth devices at single site and to extract traffic information by utilizing radio signals' temporal and spatial characteristics.

CHAPTER 3

ITSBLUE FRAMEWORK OVERVIEW

The objective of itsBlue framework is to provide ITS applications with required traffic-related information in an efficient and cost-effective way. itsBlue framework collects and manipulates road users data to generate traffic information to be consumed by ITS applications. These operations, data collection, traffic information extraction, provision and consumption constitute the main work phases of itsBlue framework. Thus, itsBlue framework is designed in multiple layers, every layer performs a part of these operations. This design reduces framework architecture complexity, and enhances interoperability and scalability. Figure 6 lists itsBlue framework layers and briefly describes their functions.

Layer	Function	Example
Application	itsBlue information consumption and ITS services production	Vehicle queue length at signalized intersection
Traffic Information Provision	itsBlue traffic information provision via a set of APIs	Providing an ITS application with vehicle locations at a signalized intersection via corresponding API method
Advanced Road User Features Extraction	Road user traffic related Information extraction	Classifying Bluetooth device to pedestrian or vehicle
Basic Bluetooth Features Extraction	Bluetooth spatial and temporal features extraction	Extracting Bluetooth device appearance time
Data Collection	- Bluetooth and location data collection, filtering and synchronization - Bluetooth device data aggregation	Collecting Bluetooth data and synchronizing it with collection locations

FIG. 6: itsBlue Framework Layers

In fact, each itsBlue framework layer involves several hardware and software components to accomplish its tasks. In this chapter, we briefly describe itsBlue framework layers and highlight hardware and software components involved in their tasks.

itsBlue framework layers tasks are performed by three parties (FIG. 7):

1. **Data collection units:** Perform data collection layer tasks.
2. **Central computing unit:** Performs the tasks of basic Bluetooth features extraction layer, advanced road user features extraction layer and traffic information provision layer. In addition, it receives, aggregates and stores collected data, which are data collection layer tasks.
3. **ITS applications:** perform ITS applications layer tasks.

As seen on figure 8, each of above parties is powered by several software modules. Figure 8 highlights every layer software modules with layer color on figure 6. The following subsections describe each layer tasks and components.

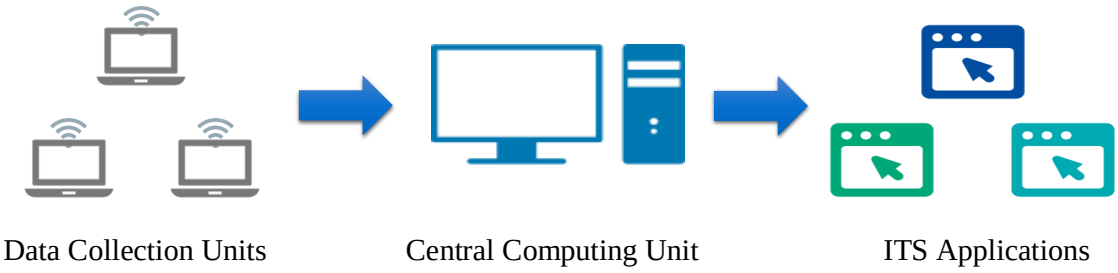


FIG. 7: itsBlue Main Components

3.1 DATA COLLECTION LAYER

In itsBlue framework, data collection is carried out by compact computing units called BlueCollect units. The BlueCollect units are placed on road infrastructure or carried on vehicles that are roving target area, and continuously collecting data. The collected data is mainly Bluetooth data and location data. Besides data collection, BlueCollect unit processes, and delivers collected data to BlueEngine, the central computing unit. BlueEngine receives and aggregates data collected by BlueCollect units, and stores it on the database.

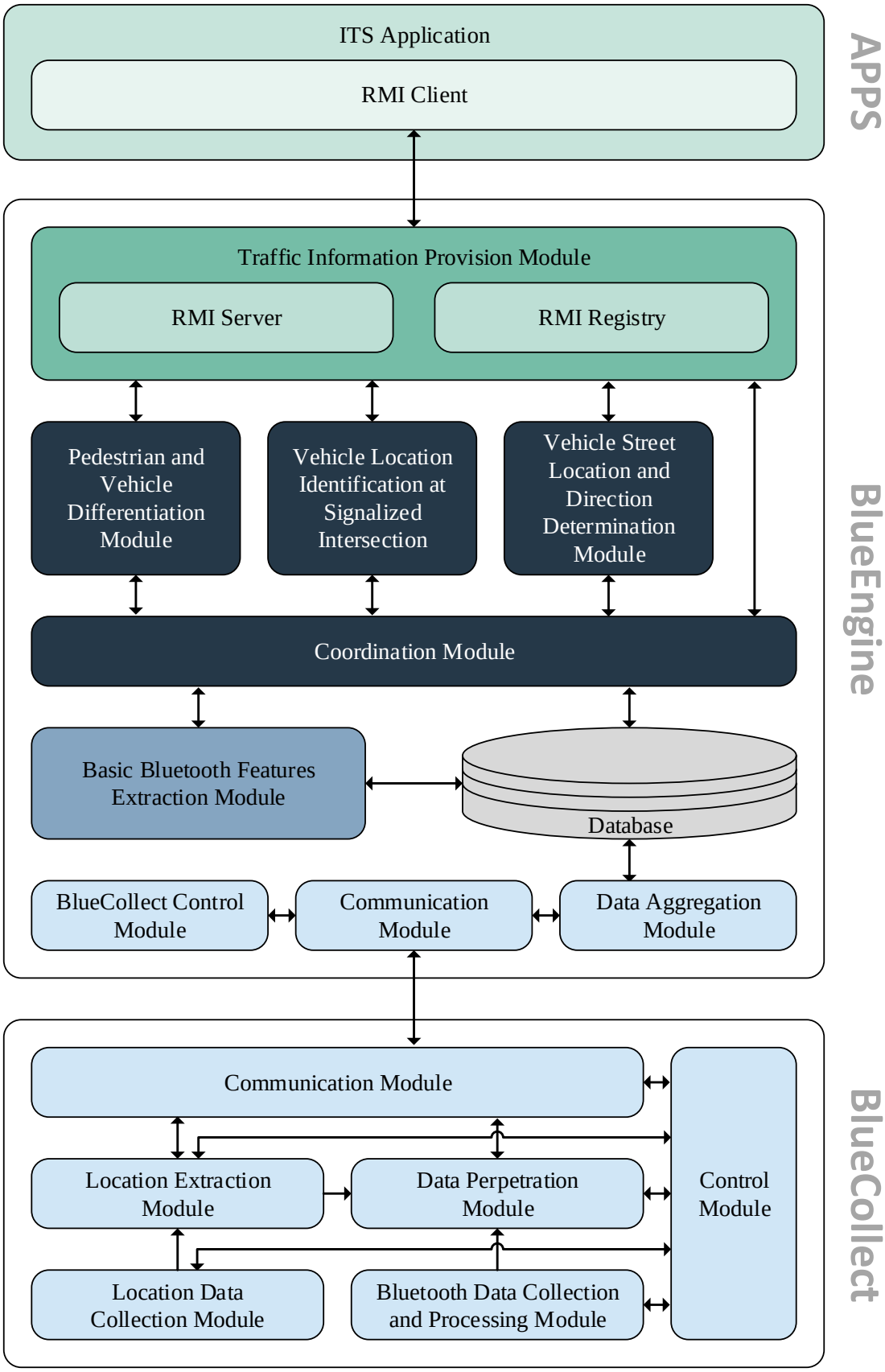


FIG. 8: itsBlue Framework Architecture

Data collection layer tasks are carried out by the following software modules:

- **BlueCollect unit side:**
  - **Bluetooth Data Collection and Processing Module:** Continuously receives Bluetooth data from road users. Obtained Bluetooth data is filtered to remove radio signal outliers. Then, transferred to Data Preparation Module. This process is repeated on time basis (*i.e. 10.24s by default*).
  - **Location Data Collection Module:** Continuously receives GPS coordinates of BlueCollect unit. Received data is transferred to Location Extraction Module on time basis (set accordingly to Bluetooth data transfer time basis).
  - **Location Extraction Module:** Uses GPS data and target area map data that requested through Communication Module to determine mobile BlueCollect unit carrier street location and direction. Determined street location and direction of BlueCollect unit, and raw GPS data are forwarded to Data preparation Module and Control Module.
  - **Data Preparation Module:** Aggregates and synchronizes collected Bluetooth and location data. Then, transfers it to Communication Module upon Control Module command.
  - **Control Module:** Supervises data collection process. Control Module initiates data collection process and organizes data transmission among BlueCollect unit modules based on temporal and spatial triggers. In addition, it executes BlueEngine commands such as BlueCollect units time synchronization.
  - **Communication Module:** Handles communications between BlueCollect unit and other parties.
- **BlueEngine side:**
  - **Communication Module:** Facilitates communication with BlueCollect units through TLS over TCP/IP to ensure system security and reliability.
  - **Data Aggregation Module:** Aggregates every Bluetooth device data and stores it on the database.

– **BlueCollect Control Module:** Maintains BlueCollect units lookup table which encompasses BlueCollect units IDs and attributes. Besides, it is responsible for data collection process initiation and termination, and BlueCollect units time synchronization. Further, it is allowed to change data collection settings like Bluetooth data collection cycle time.

3.2 BASIC BLUETOOTH FEATURES EXTRACTION LAYER

Basic Bluetooth features are extracted by the **Bluetooth Features Extraction Module**. This module extract a set of road user basic Bluetooth features (*e.g. appearance time, RSS Variance, ...*) by applying certain arithmetic operations on a range of Bluetooth data specified by location and time boarders. The basic road user's Bluetooth device features are:

- Strongest RSS
- Weakest RSS
- Median RSS
- RSS mean
- RSS variance
- Appearance time
- Number of received Bluetooth discovery response messages

3.3 ADVANCED ROAD USER FEATURES EXTRACTION LAYER

In this layer, itsBlue framework utilizes Bluetooth radio signals spatial and temporal characteristics, and the obtained basic Bluetooth features to extract traffic related information. This layer tasks are performed by the following five software modules:

- **Coordination Module:** Obtains the basic Bluetooth features or database data requested by layer modules, and transfers it to proper module. In addition, it handles data exchange between layer modules.

- **Vehicle and Pedestrian Differentiation Module:** Classifies road users detected on certain location to pedestrians and vehicle riders. This module employs machine learning technologies to classify road users relying on variances in Bluetooth feature readings.

- **Vehicle Location Identification at Signalized Intersection Module:** Identifies stopping spots of vehicles on traffic light controlled intersection. This module obtains Bluetooth radio signals received from vehicles at intersection form the database through the coordination module. Then, applies a RF sensing technology to identify vehicles locations depending on Bluetooth radio signals spatial features. Also, this module requests vehicle Bluetooth addresses from the Vehicle and Pedestrian Differentiation Module via the coordination module to avoid pedestrians positioning.

- **Vehicle Street Location and Direction Determination Module:** Utilizes vehicle basic Bluetooth features and location data of mobile BlueCollect unit carrier to determine detected vehicle street locations and directions. To do so, two kinds of data are requested through the Coordination Module. Vehicles detected on target area and their basic Bluetooth features from Basic Bluetooth Features Extraction Module. And location data of mobile BlueCollect units detected them from the database. This module utilizes obtained data to determine detected vehicles street locations and directions according to mobile BlueCollect units locations and directions during data collection.

3.4 TRAFFIC INFORMATION PROVISION LAYER

This layer facilitates the delivery of extracted road user features and traffic information to ITS applications. This layer tasks are accomplished by the **Traffic Information Provision Module**. This module implements a set of APIs that allow ITS application to obtain required traffic information of desired location and time. The APIs are implemented using Java Remote Method Invocation (Java RMI) which facilitates building distributed systems using Client / Server concept. The Traffic Information Provision Module includes the Java RMI server and registry. The RMI server implements API remote interfaces, whereas the RMI registry publicizes them. Traffic information provided by this module includes:

- Road user context.
- Vehicle locations at signalized Intersection.
- Moving vehicle locations and directions.
- Location data of mobile BlueCollect unit.
- Road user raw Bluetooth data.

The Traffic Information Provision Module requests advanced Bluetooth features from corresponding modules, and receives requested raw data from the database via the Coordination Module.

3.5 APPLICATION LAYER

The application layer is where ITS applications live. ITS applications implements Java RMI client to connect itsBlue framework, lookup required information API and invoke it.

CHAPTER 4

DATA COLLECTION

In this chapter, we thoroughly present data collection in the itsBlue framework. First, we describe the data collection infrastructure hardware and software components. Then, we discuss the advantages of Bluetooth in data collection and explain the Bluetooth communication protocol that we exploited to collect road user Bluetooth data. Next, we describe the Bluetooth data filtering approach. And, on the last section, we explain the kinds of collected data and the spatial and temporal segmentation of data.

4.1 DATA COLLECTION INFRASTRUCTURE

Mainly, the itsBlue framework is intended to collect Bluetooth data from road user devices in its vicinity in order to use the data in extracting traffic-related information. To facilitate data collection, we designed and built BlueCollect, a lightweight computing unit equipped with appropriate peripherals and powered by battery packs provided with an additional solar panel charger (FIG. 9). The compact design of the BlueCollect unit allows to deploy it in two modes:

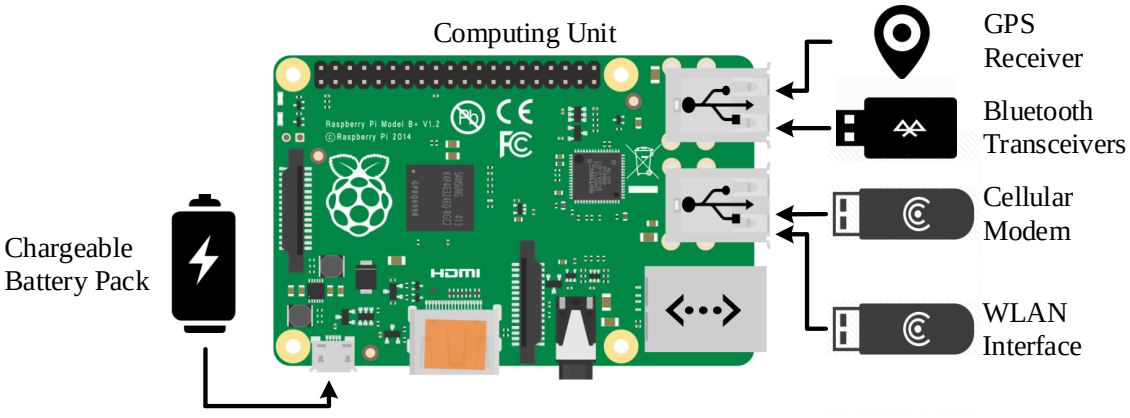


FIG. 9: BlueCollect Unit Hardware Components

1. Stationary: Where it is placed on target area infrastructure (*e.g. traffic lights, road signs, light poles, etc.*).

2. Mobile: Where it is carried on a vehicle roving the target area.

A number of BlueCollect units are distributed on a target area to obtain the required data. The data collected using the BlueCollect unit is mainly of two kinds: Bluetooth data collected from road users' devices and BlueCollect unit location data. In order to be able to collect the required data, process it, and transfer it to BlueEngine (*i.e. the central computing unit*), the BlueCollect unit encompasses the following components:

1. **Bluetooth Transceiver:** One or more USB Bluetooth adapters to collect road user Bluetooth data. BlueCollect is equipped with a Sena Parani UD100 Class 1 Bluetooth industrial adapter [18] that features up to a $1000m$ coverage range.

2. **Global Positioning System (GPS) Receiver:** A GPS unit (*i.e. NovAtel FlexPak6* [42]) to identify data collection position in mobile operating mode. The NovAtel FlexPak6 is equipped with a Satellite-Based Augmentation System (SBAS) [43] signal receiver, which narrows the location Root Mean Square Error (RMSE) to $0.6m$

3. **Wireless Local Area Network (WLAN) Communication Adapter:** A USB WLAN adapter for data exchange with the BlueEngine.

4. **Cellular Modem:** A USB cellular modem that features a high speed connection to facilitate data exchange with the BlueEngine in absence of the WLAN coverage.

5. **Computing Unit:** A credit card-sized computer (*i.e. Raspberry Pi 2 Model B* [44]) to carry out data collection, processing, and the delivery process. The computing unit synchronizes processed Bluetooth and GPS data and then transfers it to the BlueEngine. This process is performed by the following modules (FIG. 8):

   (a) **Bluetooth Data Collection and Processing Module:** Responsible for handling Bluetooth transceiver operations (*e.g. Bluetooth command fetching, error handling, etc.*) and collected Bluetooth data filtering. This module initiates Bluetooth data collection by commanding Bluetooth neighbors' discovery processes, in which the Bluetooth transceiver

broadcasts discovery messages and receives Bluetooth discovery response messages from neighboring Bluetooth devices. This process lasts for a specific period of time called "the Bluetooth discovery duration" [1]. At the end of the Bluetooth discovery duration, a radio signal filtering technique is applied on the received Bluetooth radio signals to remove any signal outliers. Then, it forwards the processed Bluetooth data to the Data Preparation Module. The Bluetooth Data Collection and Processing Module repeats this process continuously.

(b) **Location Data Collection Module:** Responsible for handling GPS operations (*e.g. GPS command fetching, error handling, etc.*). The Location Data Collection Module receives location coordinates every $1s$ and forwards it to the Location Extraction Module.

(c) **Location Extraction Module:** Responsible for obtaining BlueCollect unit street location and direction. This module utilizes the GPS traces and a mapping service to extract the BlueCollect unit street location and direction. Then it transfers the obtained street location and direction, along with the GPS traces, to the Data Preparation Module.

(d) **Data Preparation Module:** Responsible for Bluetooth and location data synchronization and transmission to the BlueEngine. The Data Preparation Module aggregates detected Bluetooth device data, associates it with a receiving location, and sorts it by reception time (FIG. 10). The prepared data is then forwarded to the Communication Module upon a Control Module command.

(e) **Control Module:** Responsible for controlling prepared data transmission to the BlueEngine and executing BlueEngine commands. Transferring collected data to the BlueEngine is driven by a number of events (*e.g. reaching a certain location, the end of Bluetooth discovery duration, etc.*). The Control Module recognizes an event's occurrence and commands the Data Preparation Module to transfer the prepared data to the BlueEngine, accordingly. In addition, the Control Module sends the BlueCollect unit information (*e.g. the BlueCollect ID, operating mode,*

[1] The default Bluetooth discovery process duration is $10.24s$. This period of time is adjustable within a range from $1.28s$ to $61.44s$

Bluetooth transceivers' information, etc.) to the BlueEngine upon initiation, and executes incoming BlueEngine commands (*e.g. data gathering initiation/termination, the data gathering cycle duration adjustment, etc.*). In addition, it synchronizes the BlueCollect unit time with the BlueEngine time.

(f) **Communication Module:** Responsible for handling the connection with the BlueEngine and exchanging data with it. The communication module provides a means of communication with the BlueEngine through Transport Layer Security (TLS) on a Transmission Control Protocol/Internet Protocol (TCP/IP) in order to ensure security and reliability.

Bluetooth Device Address: 78:47:1D:A4:B9:33
Class of Device: 0x5a020c
BlueCollect ID: SBCU-07

Receiving Date and Time	Receiving Location			RSSI	Receiver Bluetooth Address
	Coordinates	Street Segment	Direction		
2016-11-16 16:01:15.6170	36°53'15.6"N 76°18'19.2"W	49th St.	W	-63.52	00:01:95:0C:80:0D
2016-11-16 16:01:17.4980	36°53'15.3"N 76°18'24.8"W	49th St.	W	-62.27	00:01:95:0C:80:0D
2016-11-16 16:01:19.0780	36°53'15.2"N 76°18'27.6"W	49th St.	E	-81.96	88:9F:FA:EA:5C:9E
2016-11-16 16:01:20.9460	36°53'15.2"N 76°18'28.7"W	49th St.	W	-63.04	00:01:95:0C:80:0D

FIG. 10: Road User Collected and Processed Data

4.2 BLUETOOTH DATA COLLECTION PROCESS

The simplicity of Bluetooth standards expedites collecting Bluetooth data from devices in the vicinity. Bluetooth standards include a neighboring Bluetooth device discovery procedure in which a Bluetooth device searches for nearby Bluetooth devices by broadcasting Bluetooth discovery messages, while Bluetooth visible devices within range, upon receiving a discovery message, are forced to reply with a discovery response message. Collecting Bluetooth data using discovery process is advantageous compared to collecting data over a traditional Bluetooth connection. The advantages of the Bluetooth discovery process in data collection are summarized in the following points:

1. A short data receiving time, due to the absence of a connection establishment process.

TABLE 2: Frequency Hopping Synchronization (FHS) Packet Contents

Parameter	Description
BD_ADDR	Bluetooth Device Identifier
Page_Scan_Repetition_Mode	Specifies the supported page scan repetition mode
Page_Scan_Period_Mode	Specifies the mandatory page scan timer value
Class_of_Device	Refers to device kind
Clock_Offset	The difference between local clock and inquiring device clock
RSSI	The measurement of the power present in received radio signal in dBm (Range: –127 to +20)

2. No user involvement is required, whereas user approval is required to establish a Bluetooth connection.

3. Radio signal transmission power is fixed, which leads to obtaining comparable received transmission power values from transmitters, whereas, over a Bluetooth connection, the transmission power is adjusted by the transmitter to reduce power consumption.

According to Bluetooth standards [45], Bluetooth operates on the unlicensed 2.4 GHz band, which may lead to interference with other communicators. Therefore, Bluetooth uses the Frequency Hopping Scheme (FHS) to avoid interference. In Bluetooth 3.0 and former versions, the 2.4 GHz spectrum is divided into 79 channels and Bluetooth devices alternate among them in a random fashion.

In Bluetooth 3.0 and former versions, the discovery of nearby Bluetooth devices is called the Bluetooth inquiry process. In the Bluetooth inquiry process, an inquiring Bluetooth device (*i.e. master*) broadcasts inquiry messages and listens to inquiry response messages from in-range discoverable Bluetooth devices (*i.e. slaves*). To start the Bluetooth inquiry process, the master enters an inquiry sub-state, in which it uses the Inquiry Access Code (IAC) and the native clock to obtain the inquiry hop sequence, which is a sequence of 32 channels of the available 79 FHS channels (FIG. 11). The default inquiry process time is $10.24s$, the master stays $625\mu s$ on every channel, broadcasts inquiry messages which known as Identifier (ID) packets in $312.5\mu s$, and scans for replies on the other $312.5\mu s$ of the channel time window. Particularly, the 32 inquiry hop sequence channels are split onto two trains: A and B. The master broadcasts an ID packet on a train A channel within $312.5\mu s$, then

hops to a train B channel to broadcast another ID packet. Next, the master hops back to train A channel to scan for replies, after 312.5μs, it scans for replies on train B channel for 312.5μs. On the other hand, the slave enters inquiry scan sub-state to listen to master's ID packets. The slave stays 1.28s on every channel; however, it scans for ID packets for 11.25ms only, which is enough for a master to broadcast ID packets on one train of channels. Upon receiving an ID packet, the slave waits for 625μs, then replies with an inquiry response message, which known as Frequency Hopping Synchronization (FHS) packet. The slave's FHS packet encompasses its 48-bit Bluetooth addresses and other vital information (Table 2).

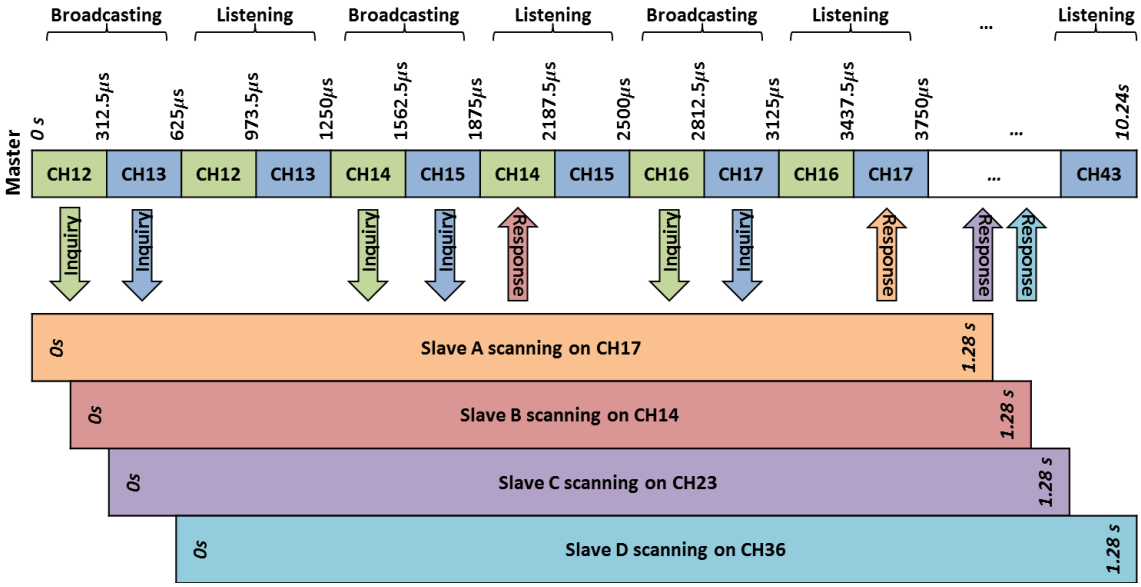


FIG. 11: Bluetooth Inquiry Process

Note: Master's green channels are train A, blue channels are train B

Furthermore, Bluetooth versions 4.0 and above received massive enhancements. The new Bluetooth modifications include improvements in the Bluetooth discovery process. Bluetooth 4.x devices operate on the same spectrum: 2402 - 2480MHz. However, the spectrum is divided to 40 channels, 3 advertising channels and 37 data channels [46]. In Bluetooth 4.x, there are two kinds of discovery procedures: Directed Advertising Events, to find known nearby devices; and Undirected Advertising Events, to find unknown nearby devices. In this research, we are interested in the undirected advertising events. In undirected advertising events, the master, which is called "advertiser" here, broadcasts the appropriate Packet Data Unit (PDU) on 3 advertisement dedicated channels (37, 38, and 39) (FIG. 12). On each channel, the

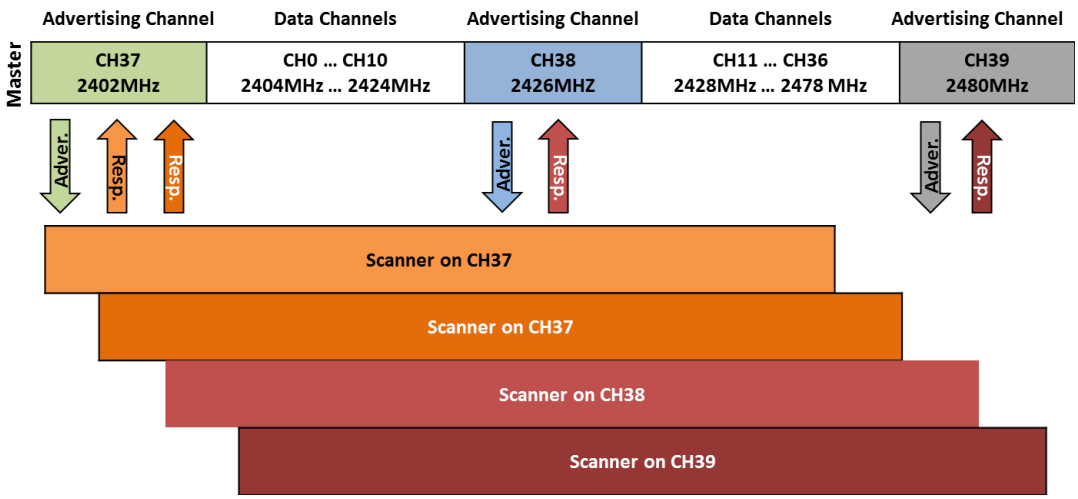


FIG. 12: Bluetooth Advertisement Process

advertiser transmits advertisement messages, and listens to responses from nearby visible Bluetooth devices. The advertisement event interval length is set by the advertiser, and it is adjustable within a range from $20ms$ to $10.24s$ [47]. On the other hand, the Bluetooth device that scans for advertisements is called a scanner instead of a slave. The scanner scans for advertisements on every advertisement channel for up to $10.24s$.

As stated above, the Bluetooth discovery process allows Bluetooth data to be collected from in-range devices in a short time, with no connection establishment or user involvement required. According to [13], five Bluetooth inquiring devices running Bluetooth version 3.0 or former are able to discover 20 visible Bluetooth devices in $3s$, whereas Bluetooth 4.x devices are able to discover a neighboring Bluetooth device within $3ms$ [14]. Furthermore, off-the-shelf Bluetooth devices are forced to reply to discovery messages with response messages via the Bluetooth stack, with no third-party application requirement.

4.3 BLUETOOTH RADIO SIGNALS FILTERING

Radio signal propagation is impacted by the environment into which the signal travels. Objects in the environment, especially metal or metal-containing objects, influence radio frequency signals in several ways, including multipath propagation and interference. For example, a transmitted radio signal may experience reflection,

refraction, scattering, or diffraction, according to the medium into which it travels and the surrounding objects, which can result in receiving the radio signal from multiple directions. This multipath propagation could either amplify or fade the received signal.

Even medium influence on radio signal propagation affects radio signals spatial and temporal characteristics, which, in turn, impacts the performance of RF sensing systems. Therefore, radio signal filtering is required, to remove the noise associated with received radio signals.

There is no radio signal filtering technique that always outperforms the others. A variety of radio signal filtering techniques showed strengths in different domains. The feedback radio signals' filtering methods showed a high performance in outdoor environments [48]. Thus, we employ the feedback approach to filter out Bluetooth radio signal outliers. In the feedback filter, the RSSI noise is removed, depending on the previously evaluated RSSI. The feedback filter is governed by this equation:

$$RSSI(n) = vRSSI(n-1) + (1-v)RSSI(n)$$

Where $0 \leqslant v \leqslant 1$. According to [48], v is set between 0.65 and 0.8.

4.4 BLUECOLLECT OPERATING MODES AND DATA COLLECTION MECHANISM

The advantageous compact design of the data collection infrastructure BlueCollect allows it to operate in a stationary or a mobile mode. The diversity of operating modes causes dissimilarity in the collected data. For instance, the amount of data collected from a moving vehicle by a stationary BlueCollect is usually less than the amount of data collected by a mobile BlueCollect moving with the vehicle. Therefore, itsBlue considers that data collected by stationary and mobile BlueCollect units are of different kinds. In order to best utilize collected data in traffic information extraction, itsBlue uses certain mechanisms to collect and process each kind of data.

4.4.1 BLUECOLLECT STATIONARY OPERATING MODE

A stationary BlueCollect unit works in a cluster of units placed together on single site to collect data in order to extract an independent single site traffic information. In fact, a cluster of stationary BlueCollect units is placed on a signalized intersection to collect road users' data in order to obtain traffic-related information (FIG. 4).

Due to its static location, a stationary BlueCollect unit is not supplied with GPS units. Thus, collected Bluetooth data is not synchronized with the location of the data collection. However, the BlueCollect unit identification number and the Bluetooth transceiver address are associated with collected Bluetooth data. Data collected by a stationary BlueCollect is transferred to the BlueEngine at the end of the Bluetooth discovery duration.

4.4.2 BLUECOLLECT MOBILE OPERATING MODE

Complete dependence upon stationary data collection units to cover the target area is expensive. Instead, we employ portable data collection units carried on vehicles moving in the target area to extend the coverage range. The itsBlue framework utilizes service vehicles (*e.g. buses, security vehicles, etc.*) that are roving the area to carry mobile BlueCollect units and to collect data from road users. To the best of our knowledge, BlueCollect is the first mobile Bluetooth-based infrastructure to collect traffic data.

As noted above, the mobile BlueCollect is continuously moving and collecting Bluetooth and location data. The collected data is divided into two kinds based on the receiving location:

1. **Street data:** The data collected while the BlueCollect unit carrier is moving on a street. The street data includes Bluetooth data and receiving street location and direction.

2. **Intersection data:** The data collected while the BlueCollect unit carrier is stopped at an intersection. The intersection data indicates the road user's appearance at the intersection at a certain time.

The mobile BlueCollect unit prepares the collected data and transfers it to the BlueEngine upon the carrier's departure of a recent street or intersection. In order to detect the BlueCollect carrier's street or intersection entry and departure, the mobile BlueCollect unit uses a two-layer target area map. The first layer is a human-readable map with street names and intersection geographic coordinates. The second layer is a graph, where edges are the streets and vertices are the intersections. Generally, a first layer street map includes more than one graph edge; thus, every graph edge corresponds to a first layer map street segment. To identify a mobile BlueCollect

unit carrier street segment, the BlueCollect unit receives the geographic coordinates of the current location via the GPS unit. The Google Maps Geocoding Service [49] is used to convert received geographic coordinates into a human-readable address. Then, BlueCollect finds the mapped street and the corresponding edges that match the given street name. To be able to recognize a carrier's exact street segment and direction, BlueCollect continuously calculates the distance between the most recent three GPS coordinates and the following two points:

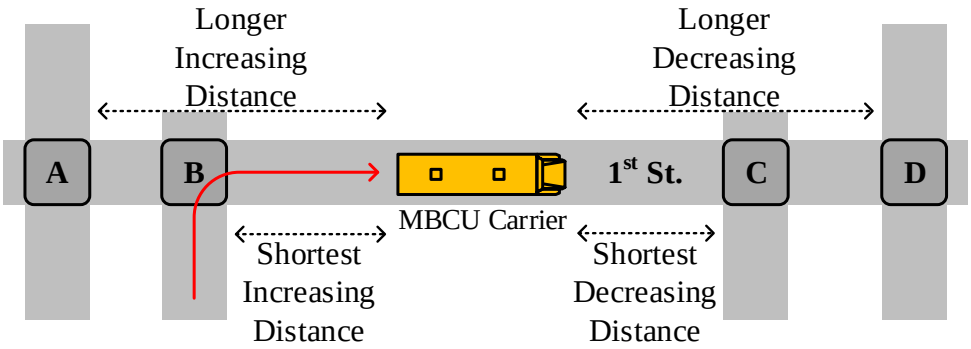


FIG. 13: Mobile BlueCollect Unit Carrier Street Segment Identification

MBCU stands for Mobile BlueCollect Unit

1. Last visited intersection geographic coordinates.

2. Possible destinations of the last visited intersection on current street according to graph layer of the map.

The Mobile BlueCollect unit selects the street segment/edge that connects the last visited intersection/vertex to the destination intersection/vertex with shortest decreasing distances from recent carrier GPS coordinates (FIG. 13). In addition, the Mobile BlueCollect carrier's U-turns are detected when the distances between recent GPS coordinates and the last-visited intersection/vertex are decreasing after a period of increase. In case the current street segment is an initial segment where the source intersection/vertex is unknown, BlueCollect calculates the distance between the recent three GPS coordinates and all of the intersections on the current street. The intersection with shortest increasing distance from recent carrier GPS coordinates is the source intersection/vertex, and the intersection/vertex with the shortest decreasing distance from the recent carrier's GPS coordinates is the destination intersection/vertex.

To be able to identify intersection entry, BlueCollect continuously calculates the distance between the recent GPS coordinates and the upcoming intersection geographic coordinates given on the map. BlueCollect recognizes street segment departure and intersection entry when the distance between the current GPS coordinates and the upcoming intersection is 15% of the street segment's length or less.

Furthermore, Mobile BlueCollect unit carrier's passengers may carry Bluetooth-enabled devices, and the BlueCollect unit may unintentionally consider these devices as vehicles. To overcome this obstacle, the mobile BlueCollect unit is supplied with additional short-range Bluetooth transceivers (*i.e.* $3m$) to detect on-board Bluetooth devices. Data from Bluetooth devices detected by these special purpose Bluetooth transceivers are discarded. To avoid discarding a vehicle mistakenly, a Bluetooth device is considered a passenger device when it replies to 30 Bluetooth discovery requests of special purpose Bluetooth transceiver.

CHAPTER 5

BASIC BLUETOOTH FEATURES EXTRACTION

On the data collection layer, BlueEngine receives data collected by stationary and mobile BlueCollect units. Then it arranges all of the detected Bluetooth device data that is collected by multiple BlueCollect units together and stores it in the database. In this chapter, we will introduce the basic Bluetooth features extraction layer, which retrieves Bluetooth device data from the database and utilizes it to extract the device's basic Bluetooth features.

5.1 BLUETOOTH FEATURES EXTRACTION

The majority of the basic Bluetooth features are extracted by applying certain arithmetical operations. Thus, we arrange every piece of detected Bluetooth device data in a set. For example, the following set, d_i, is the Bluetooth device i's collected data set.

$$d_i = \left\{ l_1, l_2, \ldots, l_a \right\}$$

Where a is the number of locations that Bluetooth device i is detected on. Let $l_j \in d_i$ where $1 \leqslant j \leqslant a$. l_j is a subset of BlueCollect units that collected Bluetooth device data at location j.

$$l_j = \left\{ u_1, u_2, \ldots, u_b \right\}$$

Where b is the number of BlueCollect units that collected Bluetooth device i data at location j. Let $u_k \in l_j$ where $1 \leqslant k \leqslant b$. u_k is a subset of Bluetooth transceivers that are plugged into BlueCollect unit k and collected Bluetooth device data.

$$u_k = \left\{ s_1, s_2, \ldots, s_c \right\}$$

Where c is the number of BlueCollect unit k Bluetooth transceivers that collected Bluetooth device data. Let $s_l \in u_k$ where $1 \leqslant l \leqslant c$. s_l is a subset of the Bluetooth RSS samples received by Bluetooth transceiver l.

$$s_l = \Big\{r_1, r_2, \ldots, r_d\Big\}$$

Where d is the number of RSS samples received by Bluetooth transceiver l.

Bluetooth signals' spatial and temporal characteristics are utilized to extract Bluetooth features. The following Bluetooth features are extracted for a Bluetooth device at a certain location:

1. **Strongest RSS:** A set of RSS values which contains the maximum RSS value of every RSS sample set obtained by every Bluetooth transceiver on the BlueCollect unit.

   $$max(u_k) = \Big\{max(s_1), max(s_2), \ldots, max(s_c)\Big\}$$

   For every $s_n \in u_k$ where $1 \leqslant n \leqslant c$,

   $$max(s_n) = r_{max}$$

2. **Weakest RSS:** A set of RSS values which contains the minimum RSS value of every RSS sample set obtained by every Bluetooth transceiver on the BlueCollect unit.

   $$min(u_k) = \Big\{min(s_1), min(s_2), \ldots, min(s_c)\Big\}$$

   For every $s_n \in u_k$ where $1 \leqslant n \leqslant c$,

   $$min(s_n) = r_{min}$$

3. **Median RSS:** A set of RSS values which contains the middle RSS value of every RSS sample set obtained by every Bluetooth transceiver on the BlueCollect unit.

   $$median(u_k) = \Big\{median(s'_1), median(s'_2), \ldots, median(s'_c)\Big\}$$

   s'_n where $1 \leqslant n \leqslant c$, is s_n ordered by RSS value. The $median(s'_n)$ yields $\tilde{x}$, which is obtained as follows:

$$\tilde{x} = \begin{cases} r_{\frac{d}{2}+1}, & \textit{if } d \textit{ is odd} \\ \frac{1}{2}\left(r_{\frac{d}{2}} + r_{\frac{d}{2}+1}\right), & \textit{if } d \textit{ is even} \end{cases}$$

4. **RSS Mean:** A set of values which contains the mean of every RSS value set obtained by every Bluetooth transceiver on the BlueCollect unit.

$$mean(u_k) = \Big\{mean(s_1), mean(s_2), \ldots, mean(s_c)\Big\}$$

$mean(s_n)$ where $1 \leqslant n \leqslant c$, is $\bar{x}$, which is obtained as follows:

$$\bar{x} = \frac{\sum_{x=1}^{c} r_x}{c}$$

5. **RSS Variance:** A set of values which contains the variance of every RSS value set obtained by every Bluetooth transceiver on the BlueCollect unit.

$$variance(u_k) = \Big\{variance(s_1), variance(s_2), \ldots, variance(s_c)\Big\}$$

$variance(s_n)$ where $1 \leqslant n \leqslant c$, is s^2, which is obtained as follows:

$$s^2 = \frac{\sum_{x=1}^{d} \left[r_x - \left(\frac{\sum_{j=1}^{d} r_j}{d}\right)\right]^2}{d-1}$$

6. **Bluetooth Device Appearance Time:** The time elapsed while the Bluetooth device's response to discovery messages sent by Bluetooth transceivers on certain BlueCollect units. Bluetooth device appearance time is obtained by calculating the difference between the receiving times of the first and last Bluetooth messages received from a device by all Bluetooth transceivers on the BlueCollect unit. To obtain a Bluetooth device appearance, let $t_{min(s_n)}$ the time of receiving first Bluetooth response message by transceiver s_n, where $1 \leqslant n \leqslant c$:

$$t_{min(s_n)} = min(t_{r_1}, t_{r_2}, \ldots, t_{r_d})$$

And let $t_{max(s_n)}$ the time of receiving last Bluetooth response message by transceiver s_n, where $1 \leqslant n \leqslant c$:

$$t_{max(s_n)} = max(t_{r_1}, t_{r_2}, \ldots, t_{r_d})$$

Hence, the Bluetooth device appearance time on a Bluetooth transceiver s_n, which is denoted as $DAT(s_n)$ is obtained as follows:

$$DAT(s_n) = t_{max(s_n)} - t_{min(s_n)}$$

The Bluetooth device appearance time is provided to the upper itsBlue framework layer with t_{min} and t_{max} in a BlueCollect unit level as follows.

$$t_{min(u_k)} = \left\{t_{min(s_1)}, t_{min(s_2)}, \ldots, t_{min(s_c)}\right\}$$

$$t_{max(u_k)} = \left\{t_{max(s_1)}, t_{max(s_2)}, \ldots, t_{max(s_c)}\right\}$$

Hence, the Bluetooth device appearance time is:

$$DAT(u_k) = \left\{DAT(s_1), DAT(s_2), \ldots, DAT(s_n)\right\}$$

7. **Number of Received Bluetooth Discovery Response Messages:** This set contains the number of Bluetooth discovery response messages received from a device by every Bluetooth transceiver on the BlueCollect unit. The number of Bluetooth discovery response messages received by transceivers on the BlueCollect unit u_k are arranged in the set $DRMC(u_k)$:

$$DRMC(u_k) = \left\{C(s_1), C(s_2), \ldots, C(s_c)\right\}$$

In addition to the device's Bluetooth features, the itsBlue framework allows ITS applications to obtain Bluetooth devices' raw data. Bluetooth devices' raw data collected by a BlueCollect unit at certain locations and times includes the following (FIG. 10):

1. Device Bluetooth address
2. Class of device
3. BlueCollect ID
4. Bluetooth response messages and related data:
    - Receiving time and date
    - Receiving location data (for data collected by a mobile BlueCollect unit)
    - RSSI
    - Bluetooth address of receiving adapter

CHAPTER 6

ADVANCED ROAD USER FEATURES EXTRACTION

This chapter is devoted to describe the novel utilization of extracted basic Bluetooth features to determine traffic information. On the advanced road user features extraction layer, the itsBlue framework classifies Bluetooth devices on the scene based on user context (*i.e. pedestrians or vehicle riders*). Next, it employs RF sensing technology to identify vehicles' stopping spots at a signalized intersection. After that, the itsBlue framework utilizes on-board Bluetooth device features and other related data to determine moving vehicles' street locations and directions.

6.1 PEDESTRIAN AND VEHICLE DIFFERENTIATION

Road user context awareness is vital to a wide range of ITS applications. For instance, pedestrian/vehicle differentiation is essential to optimizing traffic signal timing and coordination. Therefore, the itsBlue framework exploits a number of extracted basic Bluetooth features to classify road users into pedestrians or vehicle riders.

The itsBlue framework utilizes three basic Bluetooth features to differentiate between pedestrians and on-board Bluetooth devices:

1. Bluetooth device appearance time
2. Number of received Bluetooth discovery response messages
3. Bluetooth RSS variance

The Bluetooth device appearance time and the number of received Bluetooth discovery response messages are key features in revealing the context of road users. The differences in the time spent in target area between pedestrians and vehicle riders are reflected in the temporal Bluetooth features of their devices' readings. Pedestrians spend a longer time than vehicle riders in a target area most of the time, which means a longer appearance time and a higher number of received Bluetooth discovery response messages. Figures 14 and 15 show the variances in the average

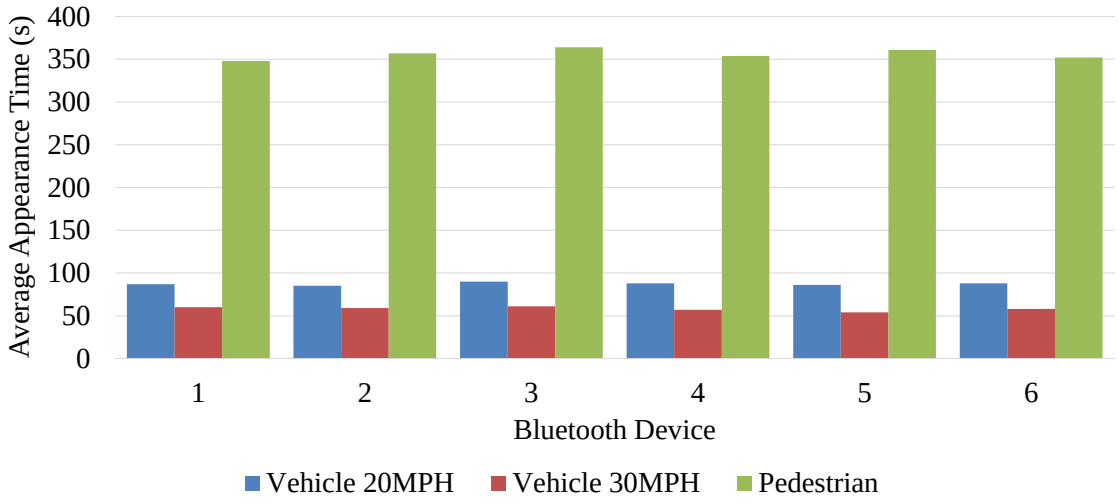


FIG. 14: Average Bluetooth Device Appearance Time

appearance times and the average numbers of received discovery response messages between six pedestrians and six vehicle riders' Bluetooth devices. During the experiment, every pedestrian or vehicle rider moved six times back and forth in a coverage range of six Bluetooth transceivers placed on sidewalks, with three on each side. As shown Figures 14 and 15, even with vehicles at low speeds of 20MPH and 30MPH, the differences in the average appearance time and the average number of received Bluetooth messages between pedestrians and on-board Bluetooth devices are clearly quite wide.

In fact, Bluetooth devices' appearance time and the number of received discovery response messages are temporal features that correlate with time spent in the target area. As long as the time spent by a vehicle and a pedestrian at an intersection vary, these features are adequate to differentiate between them. However, in certain circumstances (for example, at A congested intersection), a vehicle and a pedestrian may spend similar time on the scene, which may lead to a resemblance in their appearance times or in the number of received discovery response messages. In these scenarios, the need for a time-independent feature is raised. Therefore, itsBlue employs the Bluetooth RSS variance feature to differentiate between pedestrians and vehicle riders. In congested scenarios, vehicles are slowly moving or are making multiple stops, while pedestrians are walking at normal speeds. That variance in movement patterns leads to a prominent disparity in the variances of RSS received

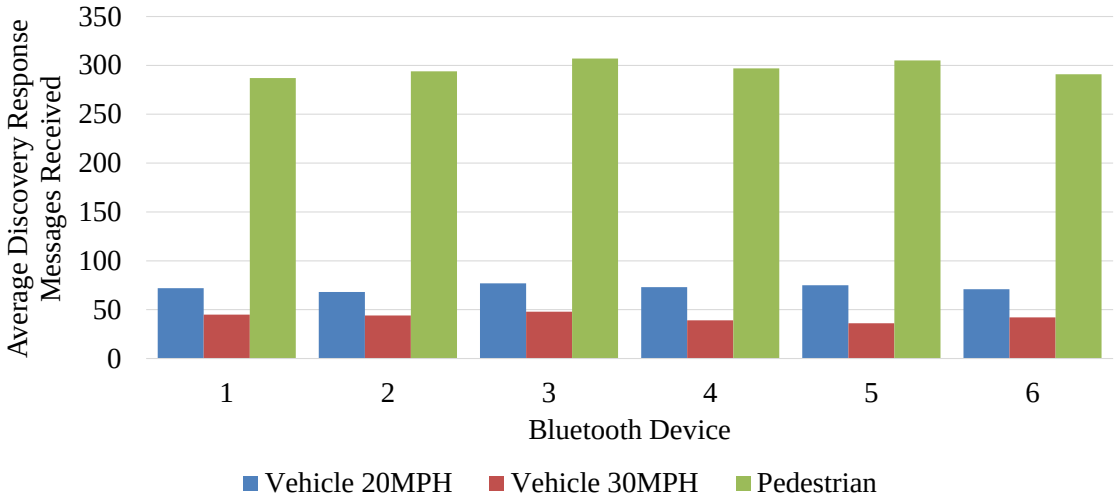


FIG. 15: Average Number of Received Messages of Bluetooth Device

from pedestrians and on-board devices. As shown in Figure 16, the difference in Bluetooth RSS variance between pedestrian devices and on-board devices confirms the ability of this feature to differentiate between pedestrians and on-board devices.

The Pedestrian and Vehicle Differentiation Module obtains the basic Bluetooth features of road users in a target area (*e.g. intersection*). The features of a Bluetooth device are provided to a BlueCollect unit level. The Pedestrian and Vehicle Differentiation Module regenerates these features for the entire target area. For instance, assume that Bluetooth device d visited location l, where b BlueCollect units are placed, and every BlueCollect unit has c Bluetooth transceivers plugged into it. The Pedestrian and Vehicle Differentiation Module generates the Bluetooth features of device d at location l as follows:

1. Bluetooth device appearance time:
   Given

$$t_{min(u_k)} = \left\{ t_{min(s_1)}, t_{min(s_2)}, \ldots, t_{min(s_c)} \right\}$$

   And

$$t_{max(u_k)} = \left\{ t_{max(s_1)}, t_{max(s_2)}, \ldots, t_{max(s_c)} \right\}$$

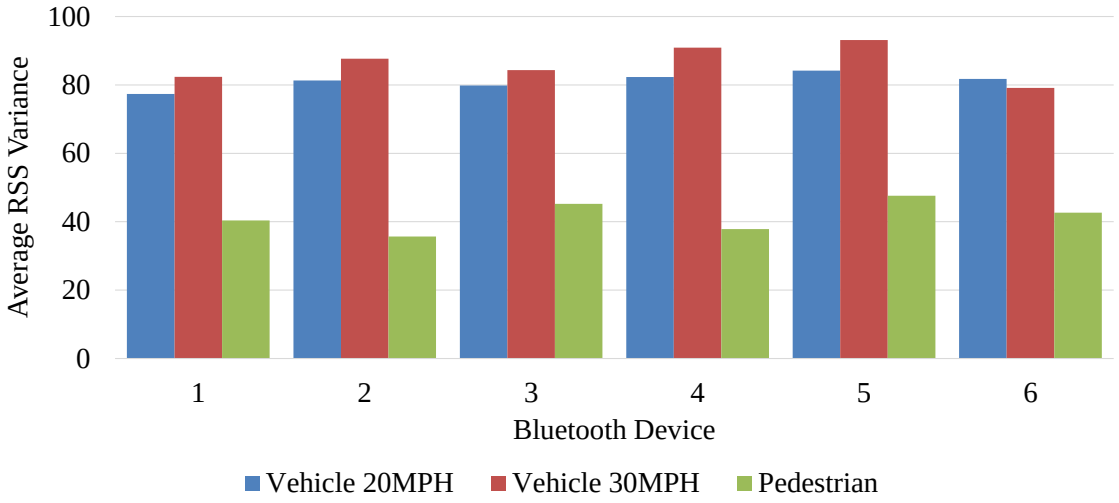


FIG. 16: Bluetooth Device RSS Variance

where $1 \leqslant k \leqslant b$. The receiving times of the most first and the most last Bluetooth response messages received by every BlueCollect unit at location l are obtained as follows:

$$t_{min(l)} = min(min(t_{min(u_1)}), min(t_{min(u_2)}), \ldots, min(t_{min(u_b)}))$$

And

$$t_{max(l)} = max(max(t_{max(u_1)}), max(t_{max(u_2)}), \ldots, max(t_{max(u_b)}))$$

The Bluetooth device d's appearance time at location l is obtained as follows:

$$DAT(d_l) = t_{max(l)} - t_{min(l)}$$

2. Number of received Bluetooth discovery response messages:
   Given

$$DRMC(d_l) = \Big\{DRMC(u_1), DRMC(u_2), \ldots, DRMC(u_b)\Big\}$$

Every $DRMC(u_n) \in DRMC(d_l)$ where $1 \leqslant n \leqslant b$, is a set of numbers of Bluetooth discovery response messages received by every Bluetooth transceiver plugged into BlueCollect unit u_n:

$$DRMC(u_n) = \Big\{C(s_1), C(s_2), \ldots, C(s_c)\Big\}$$

The number of received Bluetooth discovery response messages feature of device d at location l is obtained as follows:

$$DRMC(d_l) = \frac{\sum_{i=1}^{b} \frac{\sum_{j=1}^{c} C(s_j)}{c}}{b}$$

3. Bluetooth RSS Variance:
   Given

$$variance(d_l) = \Big\{variance(u_1), variance(u_2), \ldots, variance(u_b)\Big\}$$

Every $variance(u_n) \in variance(d_l)$ where $1 \leqslant n \leqslant b$, is a set of variance values of RSS samples received by every Bluetooth transceiver plugged into a BlueCollect unit u_n:

$$variance(u_n) = \Big\{variance(s_1), variance(s_2), ..., variance(s_c)\Big\}$$

The Bluetooth RSS variance feature of device d at location l is obtained as follows:

$$variance(d_l) = \frac{\sum_{i=1}^{b} \frac{\sum_{j=1}^{c} variance(s_j)}{c}}{b}$$

The Pedestrians and Vehicles Differentiator employs a machine learning technique to classify Bluetooth devices. Support Vector Machine (SVM) [50] and Logistic Regression (LR) [51] are commonly used classifiers. SVM and LR are discriminative classifiers, in which a training dataset is required to learn Bluetooth features, in order to use them in real time classification. In the itsBlue framework differentiator, SVM is chosen over LR for two reasons:

1. SVM requires smaller training dataset to achieve satisfactory classification accuracy

2. SVM processing time is shorter

As noted above, Pedestrian and Vehicle Differentiator deployment goes through two phases: data training and real time classification. In the data training phase, the SVM classifier is provided with the Bluetooth features of predefined pedestrians' and vehicle riders' devices. SVM validates the gained dataset using k-fold cross validation [52], in which the training dataset is divided into k parts. k-1 parts are used for data training, and the remaining part is tested against them. This operation is repeated k times to test each part once against the k-1 parts. This technique lets the classification model fit the training data as closely as possible. On the other hand, in the real time classification phase, SVM uses the aforementioned Bluetooth features of road users' devices on target areas to classify them to pedestrians and vehicle riders.

Furthermore, to enhance classification accuracy, stationary Bluetooth devices detected on the intersection are eliminated. In order to do that, a time threshold is defined. The differentiator considers a Bluetooth device stationary when its appearance time reaches the threshold. In addition, the differentiator considers any Bluetooth device that reappears on the intersection after a disappearance period as a new device. The differentiator sets a time threshold. When a previously-appeared Bluetooth device returns to the intersection, it considered as a new Bluetooth device, if the disappearance period exceeds the threshold. This process enhances classification accuracy in different cases, such when a vehicle rider's Bluetooth device is detected, and then it appears again after a while when the carrier crosses the intersection as a pedestrian.

6.2 VEHICLE LOCATION IDENTIFICATION AT SIGNALIZED INTERSECTION

Nowadays, location-based services are playing a significant role in ITS. A wide range of ITS applications rely upon location-based services. Queue length estimation at signalized intersections and origin-destination matrix generation are among these. Therefore, the itsBlue framework utilizes Bluetooth radio signals' characteristics to provide ITS applications with vehicle locations at traffic light-controlled intersections.

Bluetooth radio signals' spatial characteristics have paved the way to identifing transmitter location. Bluetooth radio signals received from a transmitter on certain spot are analogous in strength, and have a distinguishable signal strength signature.

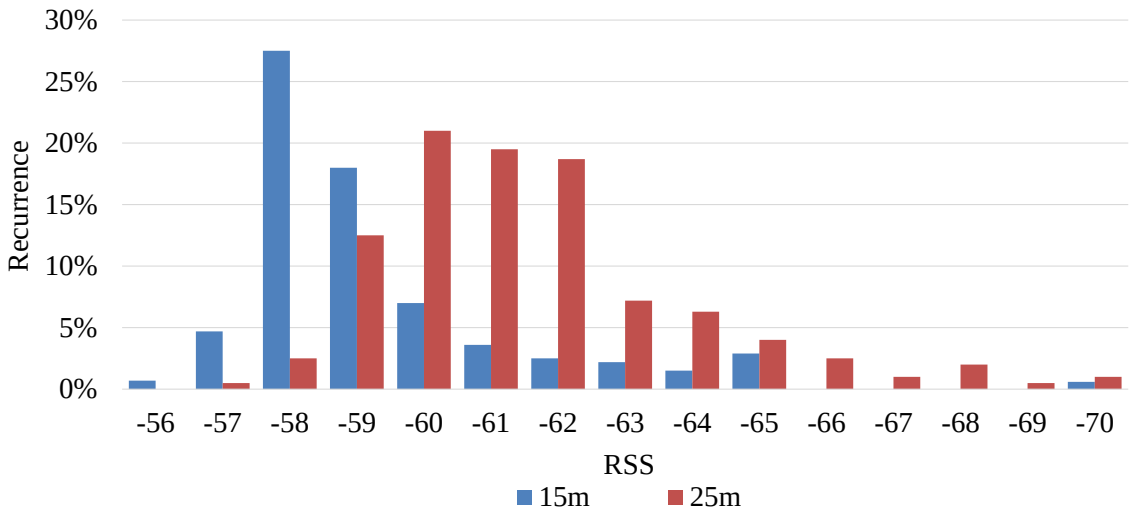


FIG. 17: Bluetooth RSS Distribution Histogram

In a simple experiment, we placed two Bluetooth-enabled cellphones at distances of $15m$ and $25m$ from a Bluetooth transceiver. As shown in Figure 17, Bluetooth radio signals obtained from each spot have distinguishable strength curves, which are called RSS distribution signatures. RSS distribution signatures of all spots create a location signal strength distribution histogram. Thus, to identify the position of a transmitting device located on one of these spots, we obtain a few Bluetooth RSS samples from it, and then we apply a probabilistic theorem to find the spot with the maximum probability, according to the location signal's strength distribution histogram.

The concept describe above is applied on vehicles at a signalized intersection in two phases: An offline phase to obtain location signal strength distribution, and an online phase to identify vehicles' locations. In the offline phase, Bluetooth RSS samples are collected from every vehicle at an intersection and are stored in the radio signal strength map database. In the online phase, a probabilistic approach is applied to identify the spot with the maximum likelihood, compared to the Bluetooth RSS samples received from vehicle. In both phases, the vehicle location identification module uses data collected by stationary BlueCollect units only. Hence, this module works on the Bluetooth transceivers level, because Bluetooth transceivers' dependency on the BlueCollect unit makes no difference in the received data. In both phases, this module receives detected Bluetooth device raw data from the database,

and associates every device's Bluetooth Address with its RSS samples. The following subsections describe in detail the offline and online phases.

6.2.1 OFFLINE PHASE: RADIO SIGNAL STRENGTH MAP CREATION:

The radio signal strength map is a database of intersection spots and corresponding RSS samples. In the offline phase, a radio signal strength map is created by storing Bluetooth RSS samples obtained from predefined devices at intersection spots. To create a radio signal strength map, we divide the target interaction into n spots (FIG. 18). Each spot is $6m \times 3m$, which are the typical dimensions of a vehicle with surrounding spaces. Then, RSS samples are obtained from predefined Bluetooth devices located at every spot. Next, the received RSS samples and their recurrences are associated with the transmitting spots and are stored on the radio signal strength map.

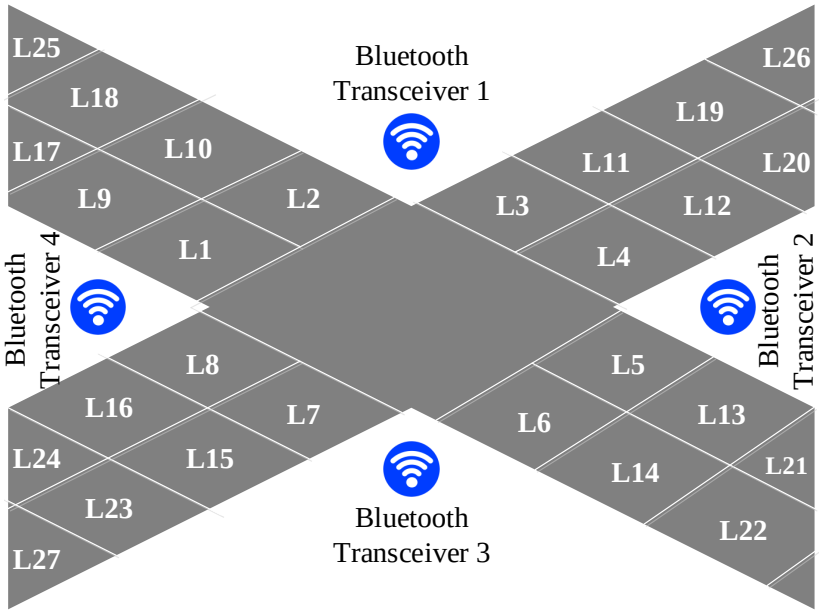


FIG. 18: Target Intersection is divided into vehicle spots

Let L_i be an intersection spot, where $1 \leqslant i \leqslant n$ and n is the number of vehicle spots on intersection L. The spot L_i's Bluetooth RSS distribution is represented by the following matrix:

$$L_i = \begin{pmatrix} P(r_{11}|L_i) & P(r_{12}|L_i) & \cdots & P(r_{1y}|L_i) \\ P(r_{21}|L_i) & P(r_{22}|L_i) & \cdots & P(r_{2y}|L_i) \\ \vdots & \vdots & \ddots & \vdots \\ P(r_{x1}|L_i) & P(r_{x2}|L_i) & \cdots & P(r_{xy}|L_i) \end{pmatrix}$$

Where r is RSS value. Every $P(r_{jk}|L_i)$ where $1 \leqslant j \leqslant x$ and $1 \leqslant k \leqslant y$, is the probability of receiving Bluetooth RSS r_{jk} from a transmitter on the spot L_i by transceiver k [53]. To obtain $P(r_{jk}|L_i)$, we divide $C_{r_{jk}}$, which is r_{jk} recurrence, over N_k, the number of RSS samples received from spot L_i by Bluetooth transceiver k.

$$P(r_{jk}|L_i) = C_{r_{jk}}/N_k$$

Then, the radio signal strength map can be expressed as:

$$M = [L_1, L_2, \ldots, L_n]$$

To avoid lane blockings, we exploit the mobility advantage of the BlueCollect unit to obtain Bluetooth RSS samples from intersection spots on-the-go. Bluetooth RSS samples are obtained from a BlueCollect unit that is carried on a vehicle visiting all of the intersection spots. The RSS sample transmitting location is obtained by the BlueCollect unit's GPS. In order to map the GPS location with the corresponding spot, the Vehicle Location Identification Module maintains a lookup table of intersection spots' coordinates. The distances between the GPS coordinates of the transmitting RSS sample and all of the intersection spots' coordinates are calculated. Then, the spot with the shortest distance from the GPS coordinates becomes the RSS transmitting spot.

6.2.2 ONLINE PHASE: VEHICLE LOCATION IDENTIFICATION:

In this phase, unknown vehicle locations are identified using Bluetooth RSS samples obtained from an on-board Bluetooth device, relying on a probabilistic approach. Figure 19 illustrates this process; the arrow colors refer to the following steps. First, stationary BlueCollect units placed on an intersection obtain RSS samples from an on-board device. For every RSS, a list of possible vehicle spots is generated using the Bayesian theorem, which finds transmitter existence likelihoods over all of the intersection spots according to the radio signal strength map (black arrows). Produced lists are aggregated and corresponding spot possibilities are gathered onto one

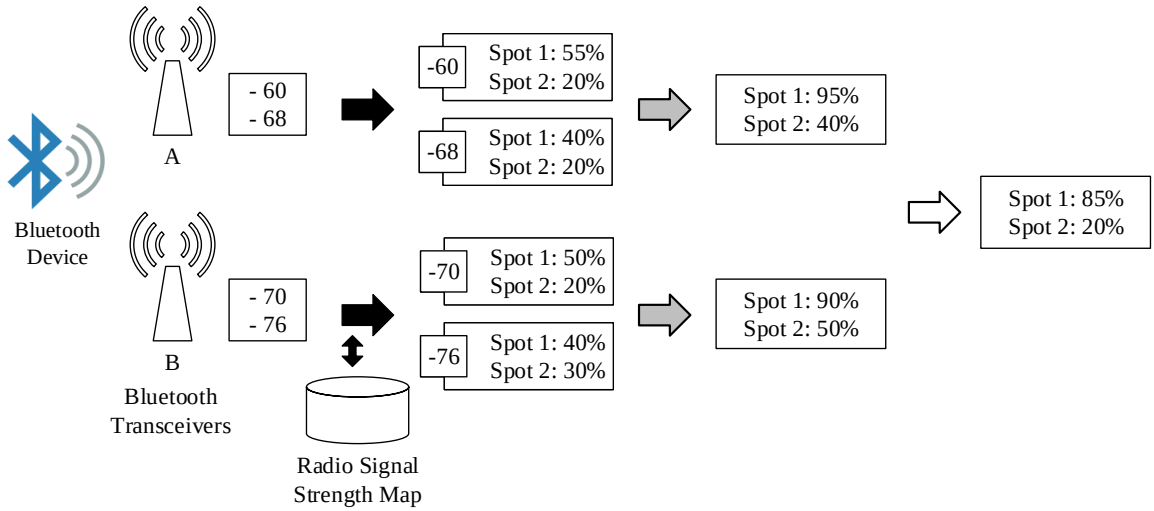


FIG. 19: Vehicle Location Identification - Online Phase

list (gray arrows). At this point, a list of possible locations is generated, using every Bluetooth transceiver RSS sample. Corresponding spot possibilities on these lists are multiplied to end up with the vehicle's final possible spots list (hollow arrow). Finally, the identifier selects the spot with the highest likelihood on the final list as vehicle's spot on intersection.

Mathematically, Bluetooth RSS samples are aggregated in matrix R:

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1y} \\ r_{21} & r_{22} & \cdots & r_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ r_{x1} & r_{x2} & \cdots & r_{xy} \end{pmatrix}$$

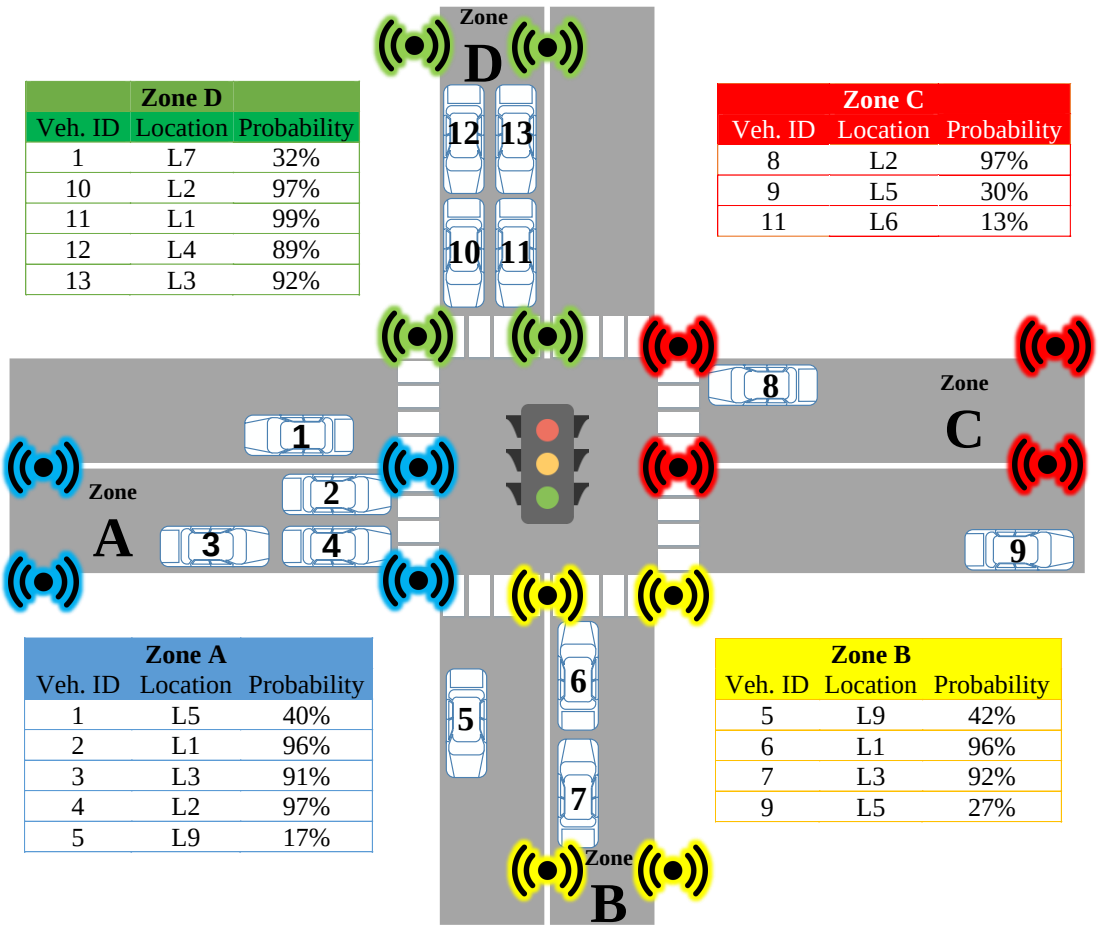
where y refers to the number of Bluetooth RSS samples, and x refers to the number of Bluetooth transceivers. To find the R matrix transmitter location l, which is the spot with the maximum probability $P(l|R)$, we apply the Bayesian theorem:

$$argmax_l[P(l|R)] = argmax_l\left[\frac{P(R|l).P(l)}{P(R)}\right] \tag{1}$$

Since $P(R)$ is constant across intersection spots, Eqn. (1) can be simplified as:

$$argmax_l[P(l|R)] = argmax_l[P(R|l).P(l)] \tag{2}$$

Note that $P(l)$ is inconstant, and that vehicle distribution over time varies widely



Zone D		
Veh. ID	Location	Probability
1	L7	32%
10	L2	97%
11	L1	99%
12	L4	89%
13	L3	92%

Zone C		
Veh. ID	Location	Probability
8	L2	97%
9	L5	30%
11	L6	13%

Zone A		
Veh. ID	Location	Probability
1	L5	40%
2	L1	96%
3	L3	91%
4	L2	97%
5	L9	17%

Zone B		
Veh. ID	Location	Probability
5	L9	42%
6	L1	96%
7	L3	92%
9	L5	27%

FIG. 20: Vehicle Location Identification Scenario

from one intersection to another. For the sake of practicality, $P(l)$ is substituted from the lookup table of the intersection vehicle distribution over time, where the vehicle distribution over a red light interval is determined. So, to calculate $P(R|l).P(l)$:

$$P(R|l).P(l) = \prod_{i=1}^{x} \left[\sum_{j=1}^{y} P(r_{ij}|l).P(l_t) \right] \tag{3}$$

Where $P(r_{ij}|l)$ is retrieved from the radio signal strength map, and t is time of receiving r_{ij} over the red light interval.

This process is applied to identify the locations of vehicles stopped at a red traffic light using a cluster of Bluetooth transceivers. The intersection is controlled by a group of Bluetooth transceiver clusters. For example, Figure 20 shows a cross intersection with four Bluetooth transceiver clusters, with each cluster controlling a zone.

In fact, itsBlue Bluetooth transceivers transmit Bluetooth discovery messages uniformly in all directions. Consequently, Bluetooth transceivers that work on a certain zone at an intersection may receive Bluetooth discovery response messages from vehicles at other zones, which may lead to locating them in a wrong zone. To overcome this obstacle, the Vehicle Location Identification Module locates only those vehicles with high likelihood. Actually, Bluetooth radio signals received from transmitters on the outer zones do not match the RSS signatures of controlled area spots with a very high likelihood. According to our experiments, the location likelihood of a vehicle on the outer zone is always less than 50%. For example, as seen in Figure 20, the Bluetooth transceivers in Zone C received RSS samples from vehicle 11 in Zone D and RSS samples from vehicle 9, which was traveling in the other direction on the street. Using Zone C transceivers' samples, the spots L5 and L6 were determined as probable spots of vehicles 9 and 11 respectively with likelihoods of less than 50%, whereas the location of the vehicle on Zone C was identified with a likelihood well above 50%. Therefore, the vehicle location identifier removed vehicles 9 and 11 from Zone C.

6.3 VEHICLE STREET SEGMENT AND DIRECTION DETERMINATION

Vehicle location and direction is another location-based service provided by itsBlue. The Vehicle Location and Direction Determination Module utilizes Bluetooth's spatial and temporal features and the awareness of the BlueCollect unit location to determine both the street segment that the detected vehicle is located on and its direction. In the following, we show how Bluetooth's spatial and temporal features can be exploited to determine the vehicle street segment and direction using data collected by mobile and stationary BlueCollect units.

6.3.1 VEHICLES DETECTED BY MOBILE BLUECOLLECT UNITS

The data collected by mobile BlueCollect units contains vehicle detection location. As seen in Chapter 4, data collected by mobile BlueCollect units is divided in two kinds:

1. **Street data:** Includes vehicle Bluetooth data and the street segment and direction of the mobile BlueCollect unit that detected the vehicle.

2. **Intersection data:** Includes vehicle Bluetooth data and the intersection on which the mobile BlueCollect unit detected the vehicle.

The Vehicle Location and Direction Determination Module utilizes the mobility advantage of the BlueCollect unit to provide ITS applications with vehicle detection streets in a target area. Such data is essential to a wide range of ITS applications. However, vehicle street and direction data provided by a mobile BlueCollect unit is the actual street segment and the direction of the BlueCollect unit that detected the vehicle. This may include misleading data, as the Mobile BlueCollect unit may detect vehicles moving on a nearby street segment or in the opposite direction. For example, the mobile BlueCollect unit described in Figure 24 detected a vehicle on a Hampton Boulevard Street while its carrier was moving on the 49^{th} Street. According to the Mobile BlueCollect unit, this vehicle was heading east on the 49^{th} Street. To avoid this obstacle, Bluetooth's temporal and spatial characteristics were exploited to determine the street segment and the direction of the detected vehicle.

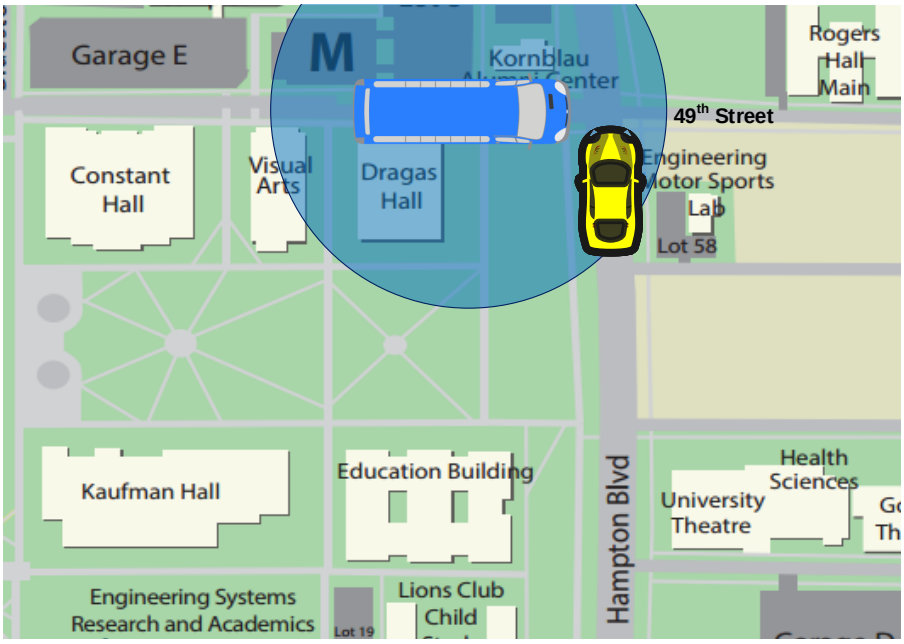


FIG. 21: Misleading vehicle location and direction provided by a mobile BlueCollect unit

The spatial and temporal features of a Bluetooth device traveling on-board a vehicle contain vital signs to its street and direction. For instance, vehicles traveling with a Mobile BlueCollect unit carrier on the same street segment can be distinguished from vehicles traveling on other nearby street segments by the number of

received Bluetooth discovery response messages. The number of messages received from a vehicle travelling with the Mobile BlueCollect unit carrier on the same street segment is higher than the number of messages received from a vehicle travelling on other street segments. Thus, the number of basic Bluetooth features that are employed to classify a detected vehicle according to its location and direction toward the mobile BlueCollect unit carrier as follows:

1. Vehicle traveled on the same street segment and direction
2. Vehicle traveled on the same street segment in opposite direction
3. Vehicle traveled on other street segments

Accordingly, to classify detected vehicles, this module obtains the following basic Bluetooth features:

1. Number of received Bluetooth discovery response messages
2. Bluetooth RSS samples mean
3. Bluetooth RSS samples variance

First, the number of received Bluetooth discovery response messages is a temporal feature that allows one to separate vehicles traveling with Mobile BlueCollect unit carriers on the same street segment and in the same direction from vehicles going in the opposite direction or on other street segments. Vehicles detected by a Mobile BlueCollect unit carrier on the same street segment and direction stay in range for a longer time than vehicles traveling in opposite directions or on nearby streets. This, in turn, leads to a jump in the number of Bluetooth messages received from vehicles traveling on the same street segment and in the same direction as the BlueCollect unit carrier, compared with others.

Second, the Bluetooth RSS mean is a spatial feature that is utilized to differentiate between vehicles on nearby streets and vehicles traveling on the same street segment as the BlueCollect unit carrier. Vehicles detected on nearby streets are distinguished by weak RSS means. The reason behind this is that these vehicles are usually located on the edge of a coverage circle, while the vehicles traveling on the same street segment with the BlueCollect unit carrier, either in the same or in the opposite

direction, are usually closer to the BlueCollect unit, which results in stronger RSS samples received.

Third, the difference in the BlueCollect unit's carrier movement direction compared to vehicle location and direction leads to a notable divergence in the variances of Bluetooth signal strength samples received from vehicles of each class in the above-mentioned vehicle location classes. The majority of Bluetooth signals received from vehicles traveling on a BlueCollect unit carrier's nearby streets are weak, due to the wide distances between these vehicles and the BlueCollect unit carrier. This stability in Bluetooth RSS samples received from these vehicles results in low variance. By contrast, the strengths of the Bluetooth radio signals received from vehicles travelling in the opposite direction of the BlueCollect unit carrier feature high variance because of the rapid change in vehicle distance from the BlueCollect unit carrier. This divergence in Bluetooth RSS variances is a key feature in classifying vehicles based on the street segment and the direction they are moving on, according to the BlueCollect unit carrier.

The Vehicle Location and Direction Determination Module receives vehicles' detection locations and basic Bluetooth features from the Coordination Module. Then, it employs the SVM to classify the vehicle according to its location and its direction, compared to the BlueCollect unit carrier. The SVM classifier is deployed in two phases: data training and real time classification. In the data training phase, the SVM classifier is provided with Bluetooth features of predefined vehicles of each class. SVM validates the gained dataset using a k-fold cross validation to let the classification model fit the training data as closely as possible. On the other hand, in the real time classification phase, the SVM classifies received detection locations and the basic Bluetooth features of vehicles in the target area to determine street segments and directions of vehicles on every street segment that vehicles are detected on by a mobile BlueCollect unit.

The street segment and direction determination is performed using street data only. The Vehicle Location and Direction Determination Module is unable to determine street segment and direction of vehicle detected by a Mobile BlueCollect unit at an intersection. At intersections, the Mobile BlueCollect unit carrier usually stops, which can result in a resemblance in Bluetooth features of all of the vehicles from all of the aforementioned location classes. Therefore, intersection data is used to indicate vehicle occurrence at an intersection with no specific street segment or

direction determined. Similarly, the street segment and direction of a vehicle that is detected while the Mobile BlueCollect unit carrier is stopped (*e.g. at bus stop*) are not determined, and that vehicle detection location data is discarded.

6.3.2 VEHICLES DETECTED BY STATIONERY BLUECOLLECT UNITS

Stationary BlueCollect units are normally used to identify vehicle locations at signalized intersections. Vehicle locations identified by the Vehicle Location Identification at Signalized Intersection Module encompass vehicle street segment and direction. Thus, the Vehicle Location and Direction Determination Module takes advantage of the availability of vehicle locations at an intersection and communicates with the Vehicle Location Identification Module through the Coordination Module in order to obtain vehicle locations at signalized intersections within the target area. Vehicles' locations at intersections are aggregated with these vehicles' locations and directions, obtained by the Vehicle Location and Direction Determination Module using Mobile BlueCollect units. Thus, vehicle occurrences at intersections provided by Mobile BlueCollect units are replaced by street segment and directions provided by the Vehicle Location Identification Module for available intersections.

CHAPTER 7

TRAFFIC INFORMATION PROVISION

In this chapter, we describe the itsBlue framework components that facilitate providing ITS applications with obtained traffic information. The itsBlue framework grants ITS application developers access to obtained traffic information via an API. The itsBlue framework's API allows ITS applications to consume the traffic information of desired time and location. In the first section, we explain the design of the itsBlue framework's API and its advantages. In the second section, we describe the set of APIs provided by the itsBlue framework to allow ITS applications to obtain required traffic information.

7.1 API DESIGN

The API is developed using Java RMI [54] to facilitate communication with ITS applications and traffic information consumption. Java RMI expedites building distributed systems using the Client/Server concept. In Java RMI, the server is responsible for implementing remote objects and for publicizing their references on the RMI registry, whereas the client (*i.e. the ITS application*) is responsible for obtaining remote references to desired objects on the server. Java RMI provides the required mechanisms for the server and the client communication and data exchange, and handles networking complications.

In order to develop a Java RMI based API, the Traffic Information Provision Module implements the following components:

- Methods Interface Definitions (*RMI Registry*): A Java class that extends remote interface and declares methods that can be remotely invoked by an ITS application.

- Methods implementations (*RMI Server*): A Java class that implements API remote interfaces. This class includes the server method, which is responsible for publicizing API methods by binding remote objects to a name in the RMI Registry.

On the other hand, the ITS application that consumes itsBlue traffic information implements a Java RMI client class, which looks up the desired service in the RMI registry and invokes it.

The following is an example of the itsBlue framework's API implementation using Java RMI. For the sake of simplicity, the implementation shows the Java RMI construction-related parts of one of the consumable methods, and abstracts others' implementation details. First, the interface definition of the remote methods:

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface itsBlueServices extends Remote {
 List<rawData> getRawData(String StartTime, String EndTime, String[]
    location) throws RemoteException; }
```

Second, the implementation of a remote method defined in the interface, which also contains the main method that creates an instance of the remote object implementation, exports the remote object, and binds that instance to a name in Java RMI registry.

```
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class itsBlueServicesImpl implements itsBlueServices {
 public itsBlueServicesImpl() throws RemoteException {}

 public List<rawData> getRawData(String startTime, String endTime, String
    [] location) {
  List<rawData> rawDataList = new ArrayList<rawData>();
  // Do Work
  return rawDataList; }

 public static void main(String args[]) {
  try {
```

```
   itsBlueServicesImpl object = new itsBlueServicesImpl();
   itsBlueServices stub = (itsBlueServices) UnicastRemoteObject.
       exportObject(object, 0);

   Registry registry = LocateRegistry.getRegistry();
   registry.bind("itsBlue", stub);

   System.err.println("Server ready");
  } catch (Exception e) {
   System.err.println("Server exception: " + e.toString());
   e.printStackTrace(); } } }
```

Third, a simple consumer that locates a remote method in the RMI registry, then invokes it.

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ITSApplication {
 private ITSApplication() {}

 public static void main(String[] args) {
  String startTime = "2014-09-10 13:07:21";
  String endTime = "2014-09-10 17:11:09";
  String[] location = new String[]{"ODU_03", "ODU_05", "ODU_07"};

  String host = (args.length < 1) ? null : args[0];
  try {
   Registry registry = LocateRegistry.getRegistry(host);
   itsBlueServices stub = (itsBlueServices) registry.lookup("itsBlue");
   List<rawData> response = stub.getRawData(startTime, endTime, location);
   System.out.println("itsBlue Traffic Information Consumed");
  } catch (Exception e) {
   System.err.println("Client exception: " + e.toString());
   e.printStackTrace(); } } }
```

7.2 SERVICE APIS

The itsBlue framework provides ITS application developers with traffic information and raw data including Bluetooth data and data collection locations. The Traffic Information Provision Module obtains the required road user features from corresponding modules, and the raw data from the database, through the Coordination Module. To enhance road user privacy, detected Bluetooth devices' addresses are replaced by a road user ID, which is a 10-digit random number mapped with the road user's Bluetooth device address on the database. Table 3 describes the provided set of APIs.

Service	**Pedestrians on Target Area**
Description	Provides developers with road user IDs and appearance times of pedestrian Bluetooth devices appearing on specified target area streets and intersections
Parameters	Start Time[1], End Time[2] and Location[3]
Return	List of locations, every location contains a list of objects, every object includes pedestrian road user IDs and appearance times
API	`list<pedestrians> getPedestrians (String startTime, String endTime, String[] location) throws RemoteException;`

[1]**Start Time**: Specifies the start time of collecting data used in information extraction. Wildcard could be thrown to start from the earliest available time.

[2]**End Time**: Specifies the end time of collecting data used in information extraction. Wildcards could be thrown to stop at the latest available time, or until termination.

[3]**Location**: The list of locations specifies collection locations for data used in information extraction. Wildcards could be thrown to include a group of locations (*e.g. ODU, which includes all ODU streets and intersections*)

Service	**Vehicles on Target Area**
Description	Provides developers with road users' IDs and with the appearance times of on-board Bluetooth devices appearing on specified target area streets and intersections
Parameters	Start Time, End Time and Location
Return	List of locations, every location contains a list of objects, every object includes vehicle road user IDs and appearance times
API	`list<vehicles> getVehicles (String startTime, String endTime, String[] location) throws RemoteException;`

Service	**Vehicle Locations at Signalized Intersection**
Description	Provides developers with vehicles' road user IDs, locations (*i.e. intersection number, street name, row number, and column number*) and appearance times at specified signalized intersections
Parameters	Start Time, End Time and Location
Return	List of locations, every location contains a list of vehicle objects, every object includes vehicle road user ID, street name, row number, column number, and vehicle existence likelihood at location and appearance time
API	`list<vehicleIntersectionLocations> getVehicleIntersectionLocations (String startTime, String endTime, String[] location) throws RemoteException;`

Service	**Vehicle Detection Locations and Directions on Target Area**
Description	Provides developers with a list of vehicles' road user IDs, street segments, directions, and numbers of received Bluetooth discovery response messages at a specified target area
Parameters	Start Time, End Time and Location
Return	List of locations, every location contains a list of vehicle objects, every object includes vehicle road user ID, street segment, direction, number of received Bluetooth discovery response messages and detecting BlueCollect unit information
API	`list<detectedVehicle> getDetectedVehicle (String startTime, String endTime, String[] location) throws RemoteException;`
Service	**Location Information**
Description	Provides developers with location information which includes: (1) Street segment info: Street segment distance and speed limits of all vehicle types (2) BlueCollect GPS locations: A series of GPS coordinates for mobile BlueCollect unit movements on specified street segments associated with time stamps
Parameters	Start Time, End Time, Location
Return	List of location objects, every object contains a street segment info and a list of mobile BlueCollect unit objects. BlueCollect unit object includes a list of GPS coordinates and time stamps
API	`list<locationInfo> getLocationInfo (String startTime, String endTime, String[] location) throws RemoteException;`

Service	**Road Uesr Raw Bluetooth Data**
Description	Provides developers with discovery response messages received in a specified target area.
Parameters	Start Time, End Time, Location
Return	List of locations, every location includes a list of road user objects, every object includes road user ID, a list of received Bluetooth discovery response messages and their receiving time stamps, and BlueCollect unit info.
API	`list<rawData> getRawData (String startTime, String endTime, String[] location) throws RemoteException;`

TABLE 3: itsBlue Framework APIs

CHAPTER 8

ITSBLUE APPLICATIONS

In this chapter, we present a number of ITS applications developed using traffic information provided by the itsBlue framework. Then, we show an itsBlue framework and applications evaluation, which includes our enhanced version of UCBT NS-2.

The first section describes a pack of intersection management applications, which provides several services such as vehicle queue length, waiting time, pedestrians' volume, etc. The second section presents the vehicle trajectories' reconstruction application. The third section is devoted to evaluation. It includes a description of our evaluation approaches and tools, in which we describe the enhanced simulation package used in evaluation. Then, we describe our ITS applications evaluation, which includes assessments and validations of traffic information provided by the itsBlue framework, in addition to a discussion of our results.

8.1 INTERSECTION MANAGEMENT APPLICATIONS

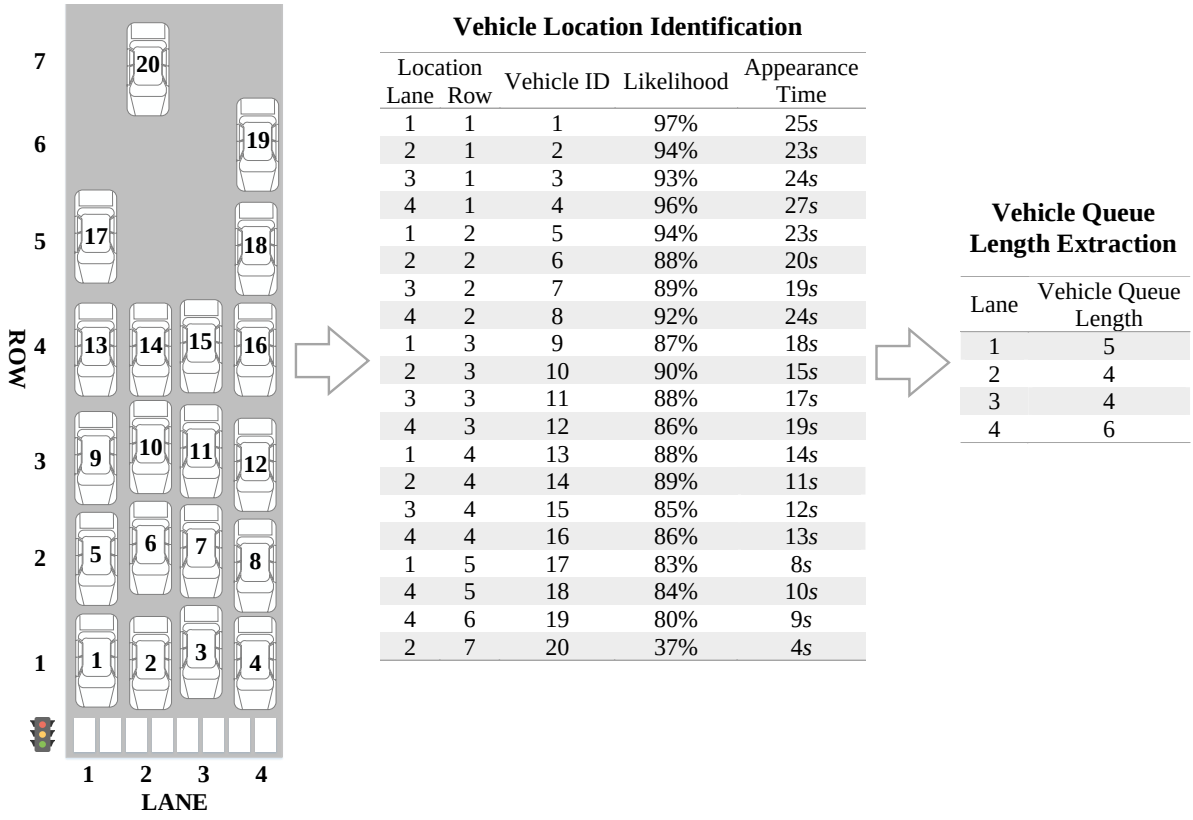
Insufficient intersection management is one of the top causes of congestion. Signalized intersection performance enhancement applications require intersection usage data and statistics. The itsBlue framework grants access to required data and provides traffic information that allows ITS researchers and engineers to develop a variety of intersection management and performance enhancement applications. For instance, the itsBlue service of showing vehicle locations at signalized intersections paves the way toward the extraction of essential information for traffic light timing optimization, such as vehicle queue lengths and vehicle waiting times.

Furthermore, the intersection management applications pack shows the ability of the itsBlue framework to extend Bluetooth utilization in ITS beyond the spatial sampling approach. The intersection management applications are provided using independent data collected by the itsBlue framework at a single site.

In the following, we describe the intersection management applications that we have developed using traffic information provided by itsBlue framework.

8.1.1 VEHICLE QUEUE LENGTH EXTRACTION

The availability of access to vehicle queue lengths at signalized intersections is essential, both for transportation agencies and for commuters. For instance, vehicle queue length is a fundamental piece of information that aids in traffic signal timing optimization which, in turn, is reflected in traffic flow smoothness and congestion alleviation. Additionally, this application is able to notify drivers of long vehicle queues in order to avoid congested routes and long delays.



Vehicle Location Identification

Location Lane	Location Row	Vehicle ID	Likelihood	Appearance Time
1	1	1	97%	25*s*
2	1	2	94%	23*s*
3	1	3	93%	24*s*
4	1	4	96%	27*s*
1	2	5	94%	23*s*
2	2	6	88%	20*s*
3	2	7	89%	19*s*
4	2	8	92%	24*s*
1	3	9	87%	18*s*
2	3	10	90%	15*s*
3	3	11	88%	17*s*
4	3	12	86%	19*s*
1	4	13	88%	14*s*
2	4	14	89%	11*s*
3	4	15	85%	12*s*
4	4	16	86%	13*s*
1	5	17	83%	8*s*
4	5	18	84%	10*s*
4	6	19	80%	9*s*
2	7	20	37%	4*s*

Vehicle Queue Length Extraction

Lane	Vehicle Queue Length
1	5
2	4
3	4
4	6

FIG. 22: Vehicle Queue Length Extraction

The vehicle queue length extractor relies on vehicle location identification at a given intersection that is provided by itsBlue framework, in order to determine the queue lengths of stopped vehicles at a red traffic light. The vehicle queue length extractor obtains vehicle locations at intersections via the itsBlue framework's API. The queue length extractor receives the vehicle locations at a target intersection, and then extracts the number of occupied locations at each lane at a red traffic light (FIG. 22). For each lane, the extractor counts the occupied spots from the first row to the row of the last identified vehicle. To avoid counting an approaching vehicle

before it reaches a stopping spot, the extractor considers only vehicles with a location likelihood of 70% and higher. According to our experiments, the stopping vehicle location likelihood is at least 70%.

8.1.2 VEHICLE WAITING TIME EXTRACTION

Vehicle waiting time is another essential service in reaching efficient intersection management. The vehicle waiting time is the time elapsed while a vehicle is stopped at red traffic light. Vehicle waiting time is another piece of information obtained from vehicles' locations at an intersection that is provided via itsBlue framework API. Vehicle waiting time is estimated by calculating the difference between vehicle occurrence and discharge times on an intersection approach. Vehicle occurrence time is the time of the receipt of the first Bluetooth discovery response message, and vehicle discharge time is the time of the receipt of the last Bluetooth discovery response message from the vehicle. The waiting time extractor filters out any vehicle with a location likelihood of less than 70%.

8.1.3 PEDESTRIANS VOLUME AND WAITING TIMES DETERMINATION

Pedestrians' volume and waiting times are significant pieces of information towards the optimization of signalized intersection and toward improvements in pedestrian facilities. In this application, target intersections' pedestrian volume is obtained from Pedestrians on Target Area, that is provided by the itsBlue framework. Pedestrian volume is the total number of detected pedestrian Bluetooth devices in the target area. On the other hand, pedestrian waiting time is the time elapsed while a pedestrian is standing on a curb waiting to cross the intersection. Actually, the pedestrian passage to the crossing zone is detected by using a short range (*i.e. 2m*) Bluetooth transceiver plugged into a stationary BlueCollect unit that is placed on a traffic light pole. To calculate pedestrian waiting time, the application retrieves road users' raw Bluetooth data from the target area by the itsBlue framework. Then, it filters out road users' IDs that are not on the pedestrian road user IDs list that was obtained from the Pedestrians on Target Area service. Next, the application uses Bluetooth response messages received by the special purpose Bluetooth transceiver to determine the pedestrian waiting time. The pedestrian waiting time is the time difference between the receipt of the first and the last discovery response messages

from the pedestrian by the short range Bluetooth transceiver.

8.2 VEHICLE TRAJECTORIES RECONSTRUCTION

The availability of vehicle trajectories is essential to draw a complete picture of traffic flow and to investigate traffic dynamics in order to improve transportation network performance. The itsBlue framework facilitates vehicle trajectory reconstruction by providing vehicle detection locations at the target area. This section is devoted to describing vehicle trajectories reconstruction using traffic information provided by the itsBlue framework.

The notion behind vehicle trajectories reconstruction is illustrated by the time lapse shown in Figure 23. On time t_1, a yellow vehicle with an on-board Bluetooth-enabled device is traveling on Hampton Boulevard from south to north, while a police patrol that is carrying a BlueCollect unit is travailing on 43^{rd} Street across Hampton Boulevard. The mobile BlueCollect unit receives Bluetooth discovery response messages from the yellow vehicle on-board device and simultaneously receives its current GPS location coordinates. On t_2, the yellow vehicle moves forward, while a bus with an on-board BlueCollect unit enters the scene. On t_3, the bus and the yellow vehicle reach the intersection of Hampton Boulevard and 49^{th} Street. The bus' BlueCollect unit receives Bluetooth discovery response messages from the yellow vehicle and receives its location coordinates from the GPS satellites. Thereafter, the vehicle trajectories reconstruction application aggregates the yellow vehicle's Bluetooth data and GPS locations received by the police vehicle and the bus to reconstruct the yellow vehicle's trajectory on Hampton Boulevard.

To apply above approach, the vehicle trajectories reconstruction application consumes the Vehicle Detection Locations and Directions on Target Area that is provided by the itsBlue framework. In this instance, the application receives a series of detection locations and times of vehicles in the target area. The vehicle detection locations series includes street segments and intersections. As seen in Section 6.3, the direction of a vehicle detected on an intersection by a mobile BlueCollect unit is unobtainable. Thus, the application aggregates vehicle-visited street segments, sorts them by detection time, and filters vehicle appearances at intersections. The obtained series of vehicle-visited street segments constitutes the initial vehicle trajectory. The initial vehicle trajectory includes visited street segments and driving directions, and the number of Bluetooth discovery response messages received from

(a) t_1 (b) t_2 (c) t_3

FIG. 23: Vehicle Trajectory Reconstruction Time Lapse Example

each vehicle on every street segment.

8.2.1 VEHICLE TRAJECTORY INCONSISTENCY RESOLUTION

In rare cases, the vehicle reconstructed trajectory may include inconsistent street segments. For example, incompatible directions on two or more street segments of a vehicle reconstructed trajectory. The inconsistency in the vehicle reconstructed trajectory may occur due to several reasons, such as a short vehicle appearance time on a street segment, because of a BlueCollect unit carrier's departure right after vehicle detection. Actually, vehicle trajectory inconsistency appears in two forms:

1. Direction incompatibility between street segments. For instance, as shown in Figure 24, the vehicle's reconstructed trajectory is suffering from a directions conflict between street segments B-F and F-G.

2. Forked reconstructed trajectory. An example: the vehicle traveled from a certain intersection to two or more street segments. For example, in Figure 24, the reconstructed trajectory is forked at intersection B to street segments B-C and B-F.

The vehicle trajectories reconstruction application resolves vehicle trajectory inconsistency, as illustrated in Figure 25. Direction incontestability is resolved by

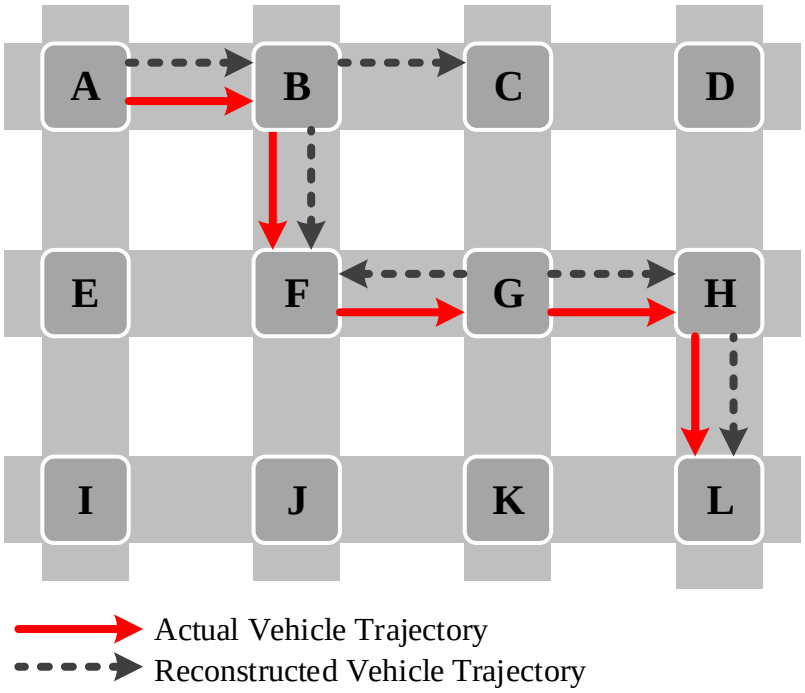


FIG. 24: Reconstructed Vehicle Trajectory Inconsistency

enforcing the direction of preceding street segments. Applying preceding street segment direction on incompatible street segment direction ensures the compatibility between vehicle movements' time lines and directions. If the preceding street segments count is one, the application applies the direction of the street segment at which itsBlue received a higher number of Bluetooth discovery response messages from the vehicle. On the other hand, a forked trajectory is resolved by removing the shortest branch. If both branch lengths are one, the application removes the street segment with lower number of received Bluetooth discovery response messages from the vehicle.

8.2.2 VEHICLE TRAJECTORY GAPS DETECTION AND FILLING

As seen above, vehicle trajectory is extracted from vehicle data that is collected by Mobile BlueCollect units carried on vehicles roving the target area or by stationary BlueCollect units placed on intersections. Because of the mobility of Mobile BlueCollect units, coverage interruptions may occur on some spots when BlueCollect units are unavailable, which can result in gaps in the reconstructed vehicle trajectory. To overcome this obstacle, we develop a vehicle trajectory gap detection and filling

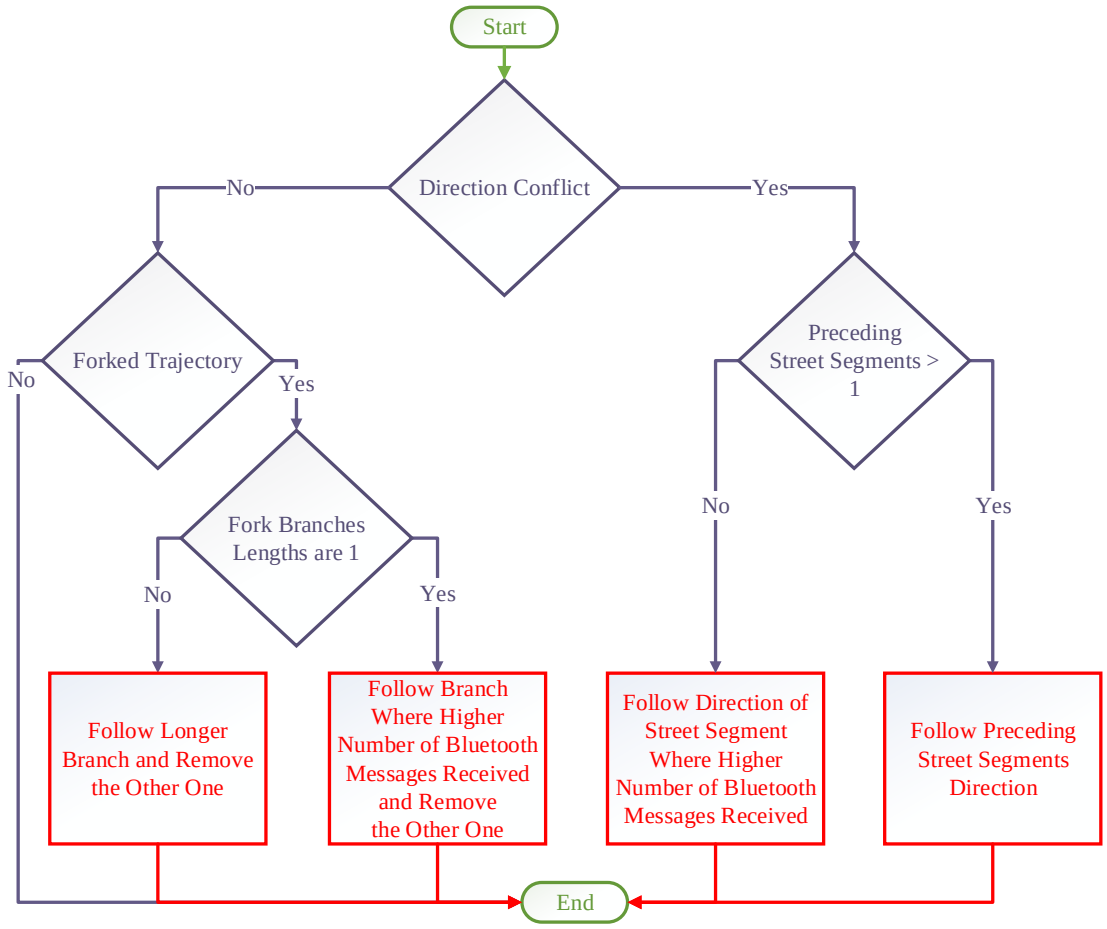


FIG. 25: Vehicle Trajectory Inconsistency Resolution Flowchart

approach.

Vehicle Trajectory Gaps Detection

To detect vehicle trajectory gaps, the vehicle trajectories reconstruction application compares extracted trajectory against a two-layer target area map similar to the one used in the mobile BlueCollect unit (Section 4.4.2). The vehicle reconstructed trajectory intersections are compared with the corresponding second layer graph vertices. Every vertex must be preceded by the source vertex; otherwise, the application identifies a trajectory gap between the recent intersection and the preceding one.

Vehicle Trajectory Gaps Filling

In order to complete the vehicle trajectory's missing part, the vehicle trajectories reconstruction application employs the Breadth First Search Algorithm (BFS) to find the path between gap bordering vertices on the target area graph. The vehicle trajectory gaps filling process involves the use of vehicle appearance at intersections that filtered out earlier in the initial vehicle trajectory reconstruction process. In the trajectory gap filling, the application looks for a vehicle's appearance at intersections in the vehicle disappearance period, which is the period of time elapsed while vehicle is traveling between gap-bordering intersections. If the vehicle is detected on any intersection during the disappearance period, the application uses the path that goes through that intersection of vehicle appearance to fill the gap. For example, in Figure 26, the initial reconstructed vehicle trajectory is broken between intersections G and E, and the vehicle is detected on intersections I and D, respectively, before it reaches intersection E. Therefore, the vehicle trajectories reconstruction application employs the BFS algorithm to find the path from intersection G to intersection I, then from intersection I to intersection D, and finally, from intersection D to intersection E.

Upon vehicle trajectory gap identification, the vehicle trajectories reconstruction application performs the following thorough steps:

1. Appoints gap bordering intersections s and d, where s_{time} $(s_t) < d_{time}$ (d_t)
2. Calculates vehicle disappearance time as $d_t - s_t$
3. Scans vehicle appearance at intersections on disappearance time
4. Sets the first found intersection of vehicle presence as d'. Note that vehicle appearances at intersections are ordered by time. Thus, intersection d' is the first intersection on which the vehicle is detected after its intersection s appearance
5. Applies the BFS algorithm to find the path between s and d'.
6. Renames d' to s and repeats the steps from 2 to 6 until it reaches intersection d.

Like the majority of graph traversal algorithms, BFS assigns weights to edges in order to evaluate path cost. In the vehicle trajectories reconstruction application, edge weight is the corresponding street segment travel time. Initially, street segment

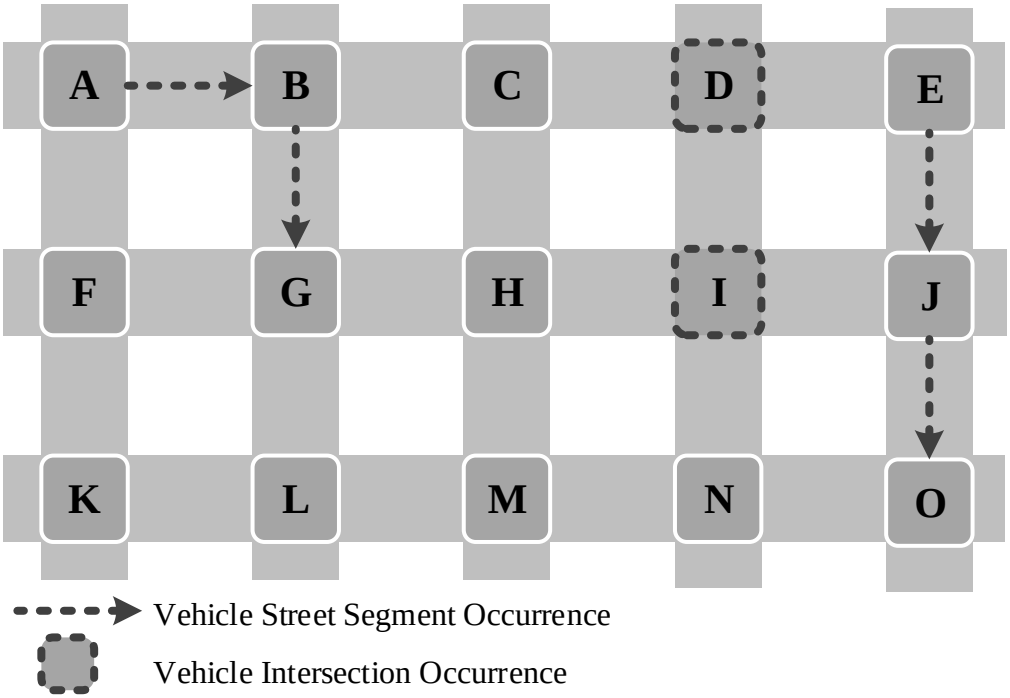


FIG. 26: The Use of Vehicle Appearance at Intersections in Filling Reconstructed Trajectory Gaps

travel time is calculated by relying on the speed limit. Afterwards, the weight is updated, based on traffic conditions. The street segment traffic condition is evaluated by relying on the Mobile BlueCollect unit's carrier travel time. The application calculates the change percentage between the mobile BlueCollect unit carrier's actual travel time and the default travel time on the street segment. Then, the street segment travel time of the normal vehicle, which is the corresponding edge weight, is updated accordingly.

In detail, the vehicle trajectories reconstruction application obtains the target street segment's normal vehicle default travel time, the Mobile BlueCollect carrier's default travel time, and the Mobile BlueCollect carrier's actual travel time by utilizing the itsBlue framework service of Location Information which provides the application with the following:

1. **Street segment information:** This encompasses the street segment speed limits of normal vehicle and the Mobile BlueCollect unit carriers, and street segment distance.

2. **Mobile BlueCollect unit carrier information:** This is a series of GPS coordinates of Mobile BlueCollect unit carrier movements on the street segments associated with the time stamps.

Upon receiving the street segment traffic information, the application calculates the following:

1. Vehicle Default Travel Time (VDT): This is obtained by dividing the street segment distance by the normal vehicle speed limit.

2. BlueCollect unit Carrier Default Travel Time (BCDT): This is obtained by dividing the street segment distance by the Mobile BlueCollect carrier's speed limit.

3. BlueCollect unit Carrier Actual Travel Time (BCAT): This is the actual time elapsed while the Mobile BlueCollect unit carrier is traveling on the street segment. BCAT is calculated by summing the time elapsed between every two GPS locations on the street segment. Further, time elapsed on stopping zones, such as bus stops, is excluded from the calculations. If the street segment includes a stopping zone, the application calculates the distance between the stopping zone and all of the BlueCollect unit GPS coordinates. If three consecutive location coordinates or more are less than $1m$ from the stop, the BlueCollect unit carrier is considered to be stopping on the stopping zone, and the location coordinates are removed from the travel time calculations.

Then, to update street segment weight, the application obtains the New up-to-date Edge Weight (NEW) as follows:

$$NEW = \left[\left(\frac{BCAT - BCDT}{BCDT}\right) \times VDT\right] + VDT$$

The BFS algorithm is used to find all of the possible paths between a given pair of vertices. Consequently, searching such a wide area in this kind of applications may impact performance. To tackle this issue, we limit the search to paths with a time similarto the time elapsed between given gap vertices. So, the application excludes any search branch that contradicts the following condition:

$$branch\ travel\ time \leqslant time\ elapsed\ on\ the\ gap + e$$

Where e is the margin of error, which is a percentage of the time elapsed on the gap. e is set depending on the traffic conditions, it goes high in low traffic and low on in high traffic areas. Once the BFS finds all of the paths that match above condition, it chooses the path with the lowest:

$$|time\ elapsed\ on\ the\ gap + e - branch\ travel\ time|$$

to fill the vehicle trajectory gap.

8.3 EVALUATION

In the evaluation section, we assess the above-mentioned itsBlue applications to show their performance and the itsBlue framework's efficiency. itsBlue applications are evaluated by conducting field experiments and by using simulation packages. Field experiments and simulation scenarios are designed to show our application's performance using a set of criteria that describe various performance aspects. Also, we validate several features extracted by the itsBlue framework to show the framework abilities to provide ITS applications with adequate traffic information. In addition, this section introduces our enhanced version of the UCBT NS-2 [55] Bluetooth simulator.

The evaluation section begins with the evaluation testbed and the simulation package. The following subsections include an itsBlue applications assessment, a related features validation, and a results discussion.

8.3.1 EVALUATION TESTBED AND SIMULATION PACKAGE

The itsBlue framework features validation and applications evaluations that are performed by conducting field experiments or by using simulation software. All field experiments are conducted outdoors at the Old Dominion University campus in Norfolk, Virginia. An experimental testbed of every experiment is illustrated in its section.

The itsBlue framework services and applications are primarily evaluated by field experiments to justify solutions' validity. However, experiment results are not adequate to show real-world performance, due to dataset limitations. Thus, a simulation package is used to assess the itsBlue applications with a large-scale dataset. In this research, we used a multiple-components simulation package. Our simulation package consists of:

1. PTV VISSIM, a microscopic multi-modal traffic simulator [56]. The PTV VISSIM is used to generate traces of a transportation network elements (*e.g. vehicles, pedestrians, traffic lights, etc.*).

2. Network Simulator 2 (NS-2) [57] with UCBT Bluetooth extension [55]. The NS-2 with UCBT Bluetooth extension is used to simulate Bluetooth communications between the itsBlue framework and the road users.

The PTV VISSIM traces of transportation network elements are generated in $200ms$ basis. The PTV VISSIM output file is converted, using a python script, into an NS-2 input file to simulate Bluetooth communications between the itsBlue framework and the road users.

In fact, the NS-2 UCBT Bluetooth extension suffers from the lack of a physical layer. Baseband is the bottom protocol layer, with each baseband packet is forwarded to the other baseband. To deliver a packet, the simulator calculates the distance between the transmitter and the receiver; if the receiver is out of the sender's range, it drops the packet. Otherwise, the packet is delivered with no received signal power consideration.

The lack of a physical layer in the NS-2 UCBT Bluetooth extension hinders the ability to obtain Bluetooth RSS, which is essential to evaluate our work. To fill the gap, we implement a physical layer sending and receiving component that utilizes the shadowing radio propagation model to obtain received signal power. The shadowing radio propagation model determines the received signal power by relying on the following equation:

$$Pr(d) = Pr(d_0) - 10\beta\ log\Big(\frac{d}{d_0}\Big) + X_{dB}$$

Where $Pr(d)$ is the received power at distance d, and d_0 is $Pr(d)$ reference point, $Pr(d)$ is calculated relatively to d_0. β is the path loss exponent, which is determined by field experiment. X_{dB} is a zero mean Gaussian random variable (measured in dB) added to reflect the variation in average received power. X_{dB} standard deviation is σ_{dB}, which is known as the shadowing deviation and is obtained by experiment [57].

To show the validity of the our enhanced NS-2 UCBT Bluetooth extension, we design a field experiment to obtain a set of Bluetooth RSS samples, in order to validate the Bluetooth RSS samples obtained from a similar simulation scenario

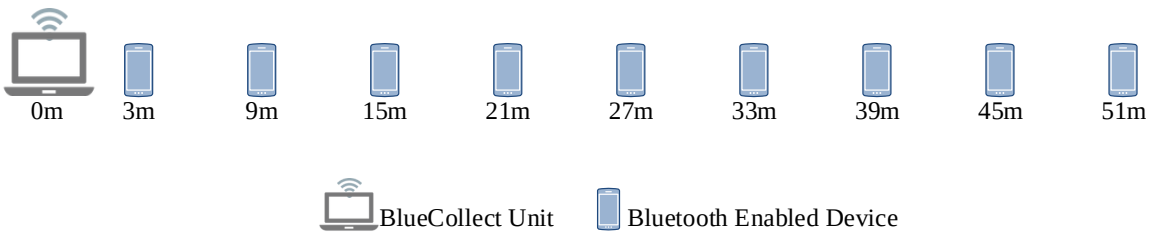


FIG. 27: Enhanced NS-2 UCBT Bluetooth Extension Validation Experiment Setup

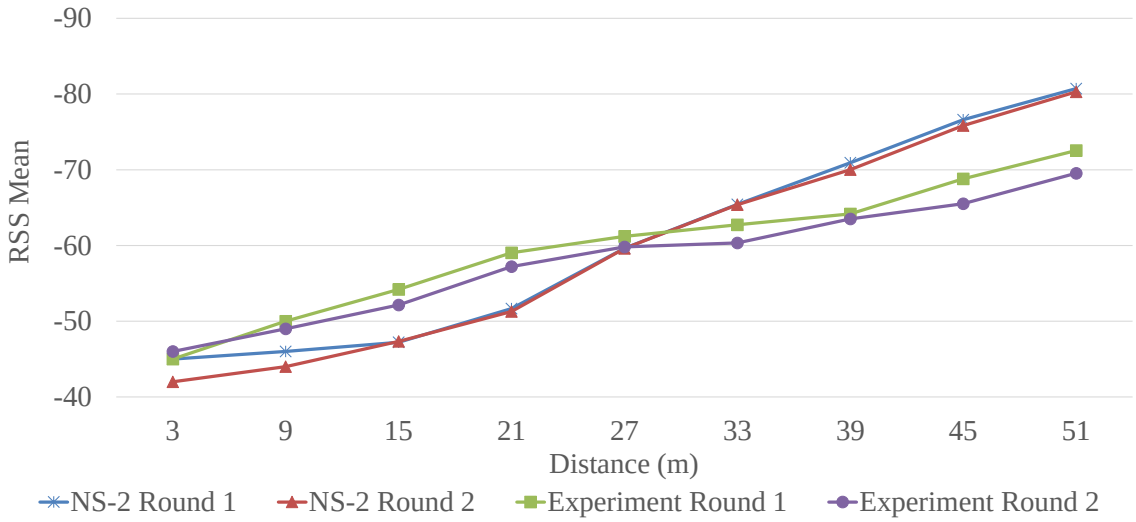


FIG. 28: Mean of RSS Obtained from Transmitters on Several Distances in Simulation and Field Experiment

against it. The experiment setup is shown in Figure 27, where nine Bluetooth devices are placed along a line, with $6m$ separating every device from its succeeding one. A BlueCollect unit is set next to them to collect data. The experiment and the corresponding simulation are performed twice. After each time, the mean of the obtained Bluetooth RSS samples is calculated for every Bluetooth device of the nine. The experiment and simulation outputs show that the behavior of the Bluetooth RSS over distance in NS-2 almost matches the experiment of the Bluetooth RSS (FIG. 28).

8.3.2 VEHICLE QUEUE LENGTH ESTIMATION EVALUATION

In this subsection, we evaluate one of the intersection management applications: the vehicle queue length extraction. The vehicle queue length extraction application is heavily dependent upon the vehicle locations identification at a signalized intersection that is provided by itsBlue framework. Thus, the vehicle locations at a signalized intersection are first evaluated. Then, we evaluate the vehicle queue length extraction application.

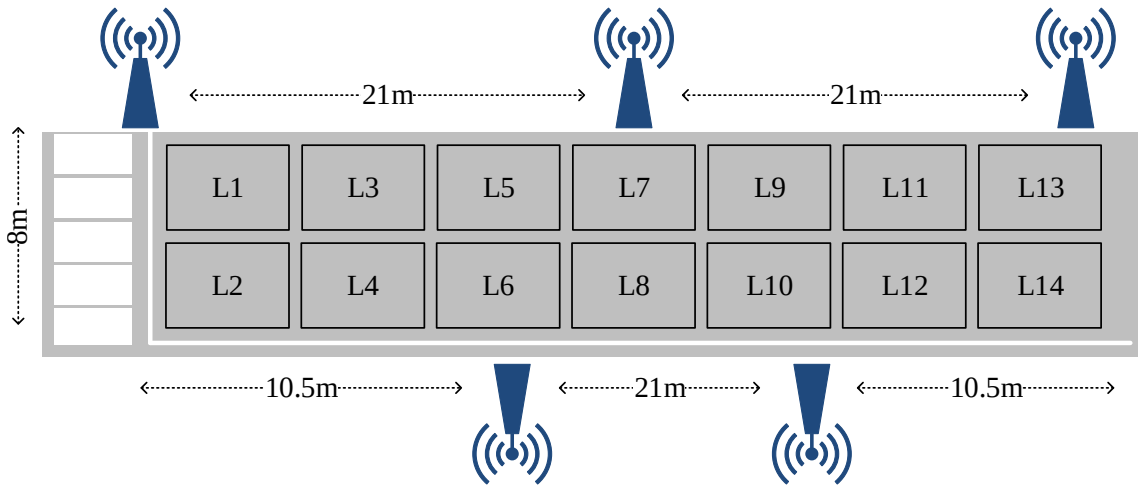


FIG. 29: Vehicle Locations Identification Evaluation Experiment Setup

A. Vehicle Locations Identification at Signalized Intersection Evaluation

The vehicle location identification is evaluated by measuring the TPR of identified vehicle locations and by studying the impact of following factors:

1. Number of used Bluetooth transceivers.

2. Number of obtained RSS samples from on-board device.

Experiment Setup

The experiment is conducted on a cross intersection-like area in parking lot 42 on the ODU campus. Two BlueCollect units with total of five Bluetooth transceivers are deployed, as seen in Figure 29. The experiment ground is divided into 14 spots, representing vehicle locations at a traffic light intersection. An on-board mobile BlueCollect unit is used as a road user Bluetooth device. The carrying vehicle stops at every spot for less than $30s$, while Bluetooth transceivers of the stationary BlueCollect units that are deployed on the sidewalks are scanning for nearby Bluetooth devices. Fifty Bluetooth discovery response messages are collected from each spot to create the area radio signal strength map. A vehicle distribution lookup table is determined by observation, and the data is collected from the intersection of Hampton Boulevard and 49^{th} Street on the ODU campus on a weekday from 3

TABLE 4: Vehicle Spots Occupancy Distribution on a Signalized Intersection Approach

Row	1	2	3	4	5	6	7
t_0 - t_9	70%	30%	0%	0%	0%	0%	0%
t_{10} - t_{19}	42%	37%	21%	0%	0%	0%	0%
t_{20} - t_{29}	28.5%	28.5%	25%	18%	0%	0%	0%
t_{30} - t_{39}	21%	21%	21%	20%	13%	4%	0%
t_{40} - t_{49}	19%	19%	19%	19%	14%	10%	0%
t_{50} - t_{59}	16%	16%	16%	16%	16%	12%	8%

P.M. to 3:45 P.M. (Table 4). Thereafter, RSS samples that are received from on-board Bluetooth-enabled smartphones are employed to identify their locations and to calculate the identified vehicle locations' TPR. The location identifier is tested 250 times, on five folds, starting with one RSS sample in the first fold. The number of RSS samples is incremented by one, respectively, for every fold. Each testing fold of the five is repeated 50 times, with new RSS samples set every time.

Result

The results show that the TPR of identified vehicle locations is very close to 100% when all five Bluetooth transceivers are used (FIG. 30). In that case, the tiny error is mostly one spot away from the actual spot (FIG. 31). The identified vehicle locations' TPR goes slightly down to around 95% when the number of RSS samples is reduced to three and one. In addition, the result shows the vehicle location identifier's ability to maintain high vehicle location identification accuracy using a combination of four or three Bluetooth transceivers, with an error of two locations away from the actual location about 95% of the times.

As seen in the experiment outcome, the itsBlue vehicle location identification service shows promising performance in the 90th percentile for TPR when three or more Bluetooth transceivers are used to identify vehicle locations, whereas the number of obtained Bluetooth RSS samples' impact is less than 10% in most cases.

B. Vehicle Queue Length Extraction Application Evaluation

The performance of vehicle queue length extraction application is assessed by the TPR of the obtained queue lengths. We calculate the TPR of the extracted vehicle

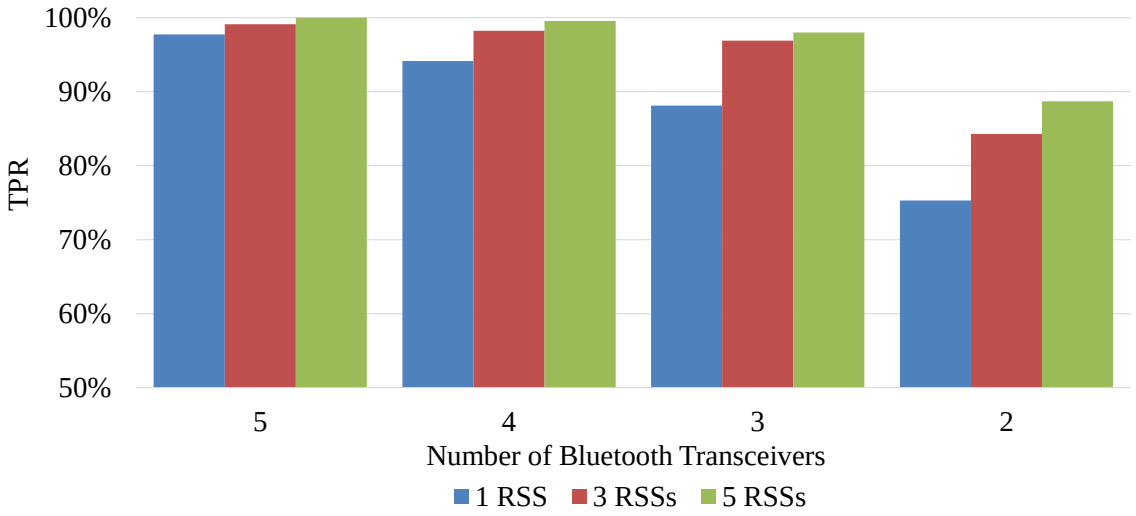


FIG. 30: Vehicle Locations Identification Accuracy

queue lengths by comparing it to the ground truth data obtained from the PTV VISSIM simulator.

Simulation Scenario

We design a simulation scenario which features a signalized cross-intersection with 400 vehicles. The simulation duration is set to $28min$, and every traffic light turns to green for $20s$ and red for $60s$. Six Bluetooth transceivers are distributed on the sides of each traffic light upstream (FIG. 32) and continuously broadcast Bluetooth discovery messages on a discovery process duration of $10.24s$ for each cycle. The Bluetooth data collected from vehicle during the first $8min$ is used to create the radio signal strength map, whereas the data collected on the remaining time is used to estimate vehicle queue lengths. Each traffic light upstream is divided into 18 spots. During the data training phase, the actual vehicle locations that are obtained from the PTV VISSIM simulator are used in the radio signal strength map creation and the intersection vehicle distribution extraction. In the real time phase, the actual vehicle locations are used as ground truth.

Result

Figure 33 shows that the TPR of the vehicle queue lengths extraction on the first Bluetooth discovery cycle is about 60% in the best case. This low vehicle queue length

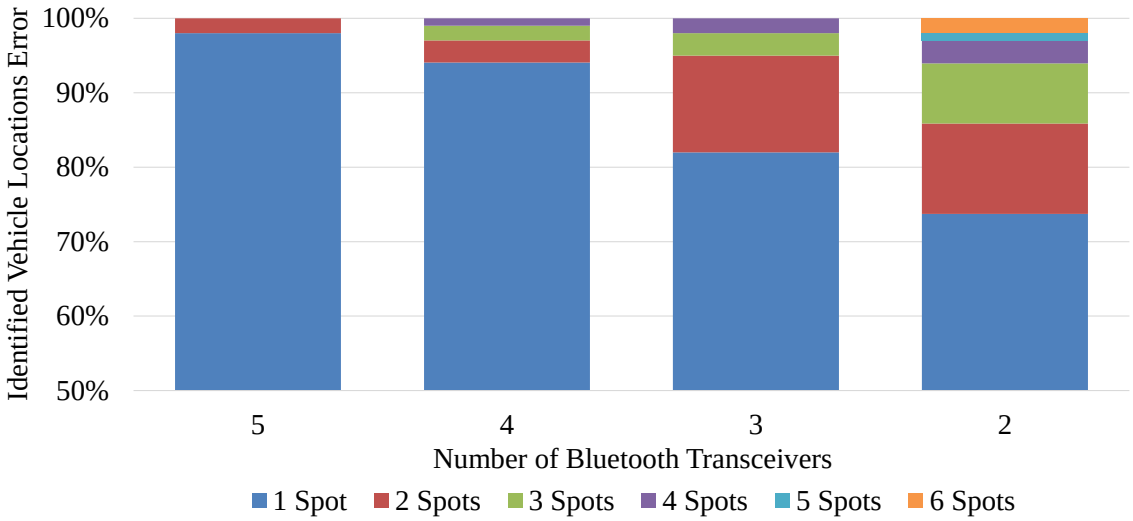


FIG. 31: Vehicle Locations Identification Error Distribution

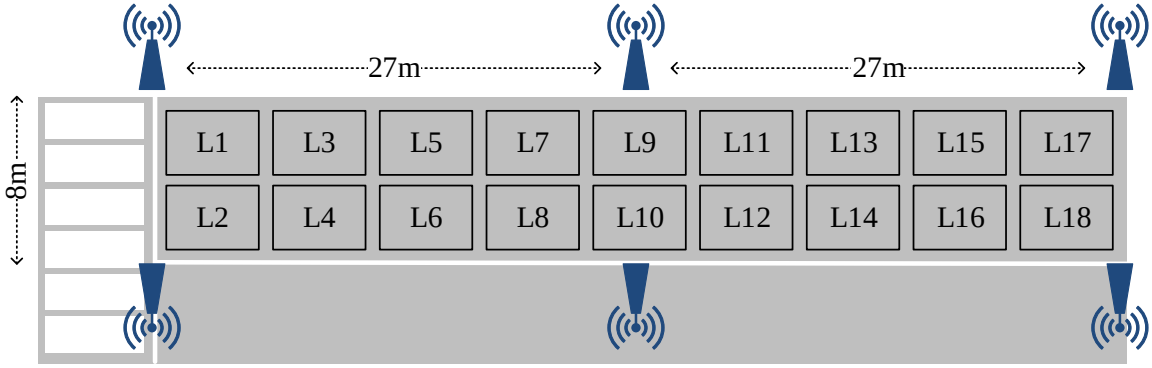


FIG. 32: The Configuration of one of traffic light approaches on the Intersection of Vehicle Queue Length Extraction Evaluation Simulation Scenario

estimation accuracy occurs because the number of RSS samples that are obtained after vehicle reaches the stopping spot is low on the first Bluetooth discovery cycle. In the following discovery cycles, the vehicle queue length estimation TPR notably increases, and reaches 96% after the fourth cycle and 98% after the fifth cycle, when all six Bluetooth transceivers are in use. The result confirms the itsBlue application's ability to accurately estimate vehicle queue length at signalized interactions using four or more Bluetooth transceivers. Despite the less than 80% vehicle queue length estimation TPR in the first two Bluetooth discovery cycles, the vehicle queue length extractor shows high performance by the third cycle, which means about 30s, whereas

the red light duration is more than $30s$ in most cases.

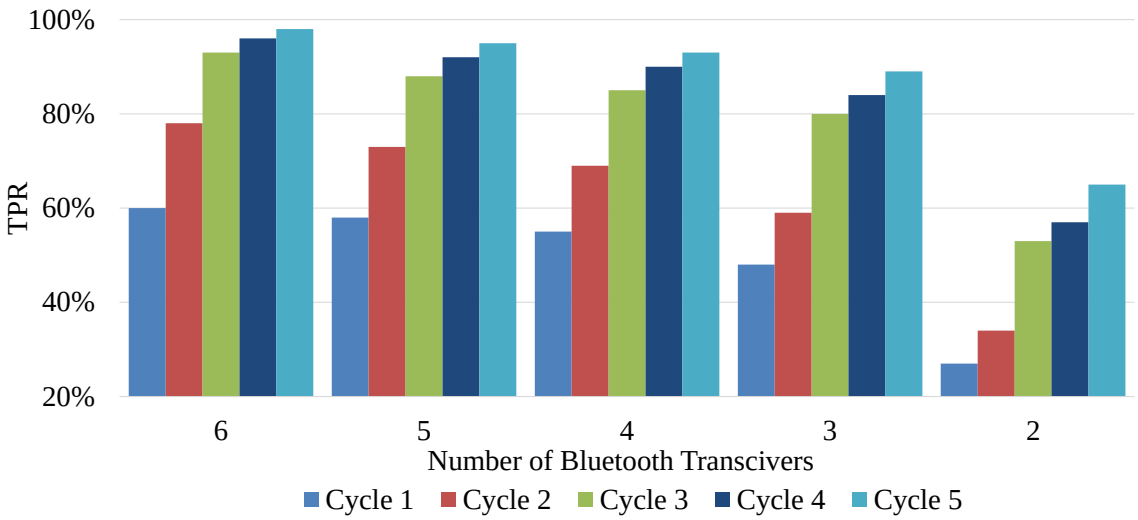


FIG. 33: Vehicle Queue Length Estimation Accuracy

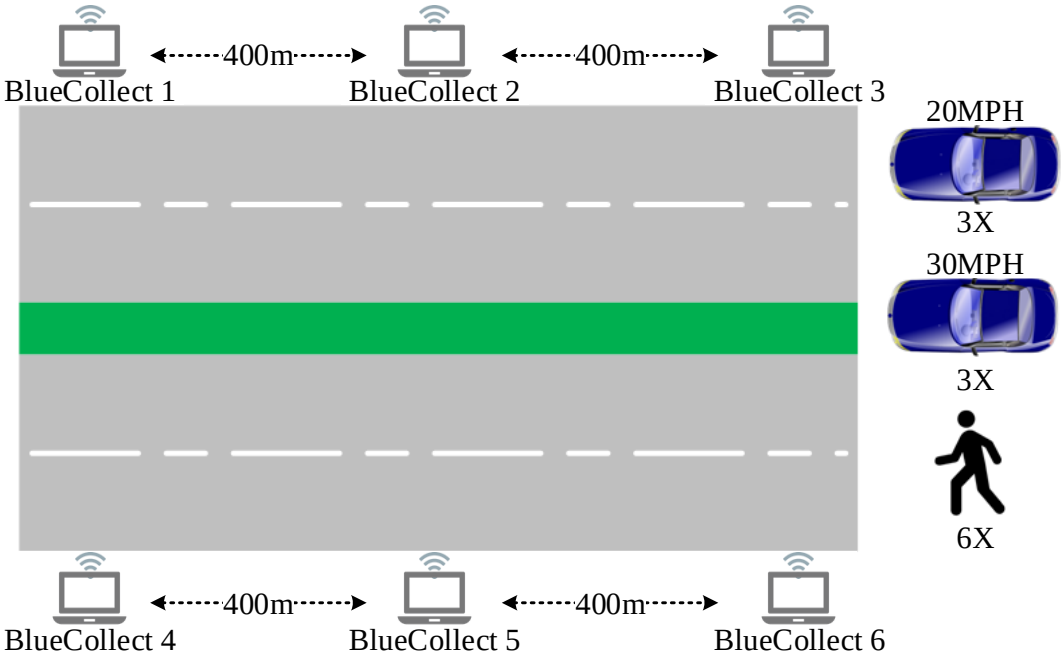


FIG. 34: Pedestrian and Vehicle Differentiation Evaluation Experiment Setup

8.3.3 PEDESTRIANS VOLUME AND WAITING TIMES DETERMINATION EVALUATION

Pedestrian volume and waiting times are extracted directly from the data provided by Pedestrians on Target Area. Therefore, in this subsection, we evaluate pedestrian and vehicle differentiation performance, which is reflected in the pedestrians' volume and waiting times' determination performance.

In this subsection, pedestrian and vehicle differentiation is evaluated three times, in a field experiment and in two simulation scenarios. The road users' classification accuracy is expressed by the TPR of classifying the pedestrians and the vehicles.

Experiment Setup

This experiment is conducted on 43^{rd} Street on the ODU campus. Six BlueCollect units are placed along sidewalks, with three on each side, separated by $8m$ (FIG. 34). Bluetooth data is collected from six Bluetooth-enabled smartphones carried by volunteers walking back and forth on the sidewalk, three times. Then, the volunteers drive vehicles back and forth at two different speeds, 20MPH and 30MPH, driving at each speed three times. The collected data is divided into two parts: the classifier

training dataset, which constitutes 10% of the collected data, and the real-time road users data, which is 90% of collected data.

Experiment Result

The result shows the vehicle classification TPR of about 80% in $3s$ (FIG. 35). On the other hand, the pedestrian classification TPR in $3s$ is slightly improved, compared to the vehicle classification TPR (FIG. 36). Pedestrian and vehicle classification TPRs reach the 90th percentiles in $15s$. Figures 35 and 36 show the correlation between the gradual TPR improvement and the Bluetooth discovery cycle time increase. Both pedestrian and vehicle classification TPRs jumped to around 95% when discovery cycle time reached $25s$.

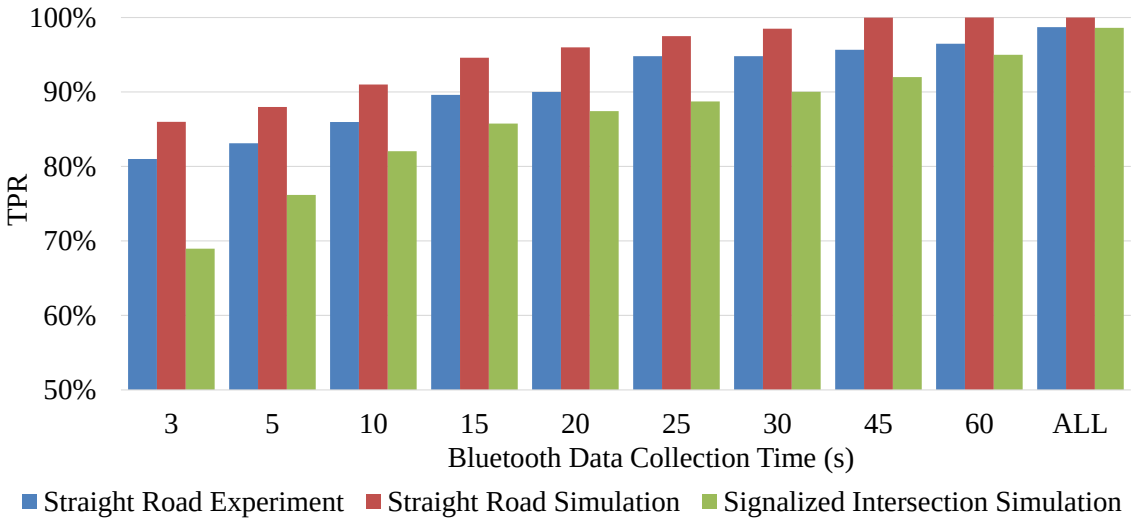


FIG. 35: Vehicle Classification Accuracy

Simulation Scenario No. 1

In this scenario, BlueCollect units are distributed in similar way to the field experiment, with $500m$ to separate them (FIG. 37). The PTV VISSIM simulator generates 4000 vehicles moving on speeds ranging from 24.9MPH to 37.3MPH, and 500 pedestrians moving on sidewalks on speeds ranging from 2.4MPH to 3.8MPH. This simulation scenario lasts for $60min$.

The simulation result shows that the differentiator achieves pedestrian and vehicle classification TPRs of 85% in $3s$, and both exceed 90% in $10s$ (FIG. 35 and FIG.

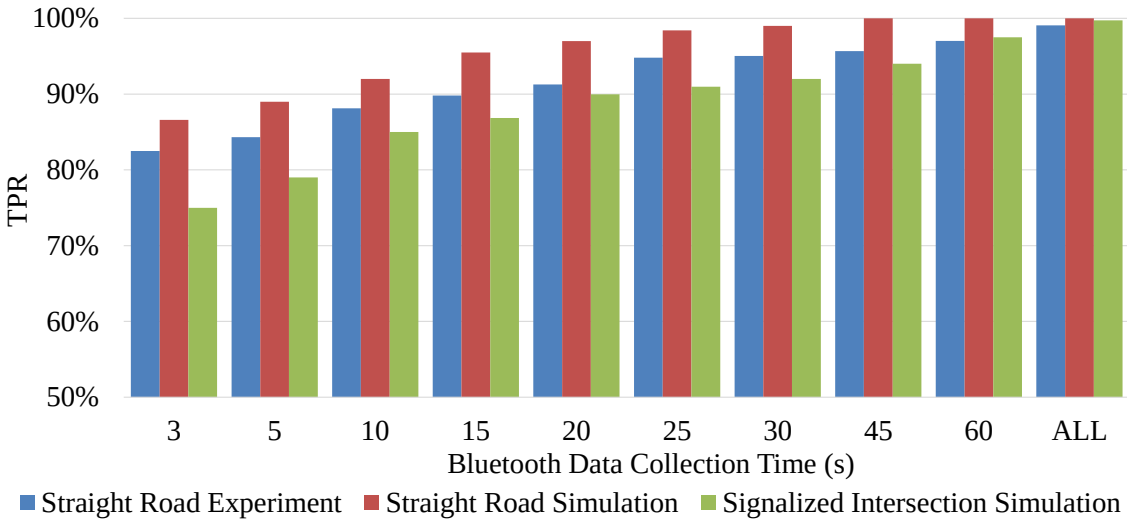


FIG. 36: Pedestrian Classification Accuracy

36). In addition, the pedestrian and the vehicle classification TPRs reach 100% when data collected in 45s is used.

Simulation Scenario No. 2

This simulation scenario is designed to evaluate the pedestrian and vehicle differentiator performance in a signalized intersection. The model includes 4000 vehicles and 500 pedestrians crossing an intersection over an hour. As seen in Figure 38, the BlueCollect units are placed on top of traffic lights.

The result shows that the TPR of classifying pedestrians exceeded 77% in 3s (FIG. 36), while the vehicle classification TPR reaches 77% in 5s (FIG. 35). The differentiator correctly classifies 90% of pedestrians, when the Bluetooth discovery cycle is adjusted up to 20s, whereas it needs a 30s Bluetooth discovery cycle duration to reach a vehicle classification TPR of 90%. The pedestrian classification TPR reaches around 99% when data collected over the entire device appearance time is used. On signalized intersections, pedestrians and vehicles appear on the scene for longer times than on straight roads, due to stops on red traffic lights which,in turn, allows the classifier to accurately identify them because of the high variance between Bluetooth features readings of pedestrians and vehicles over a long time.

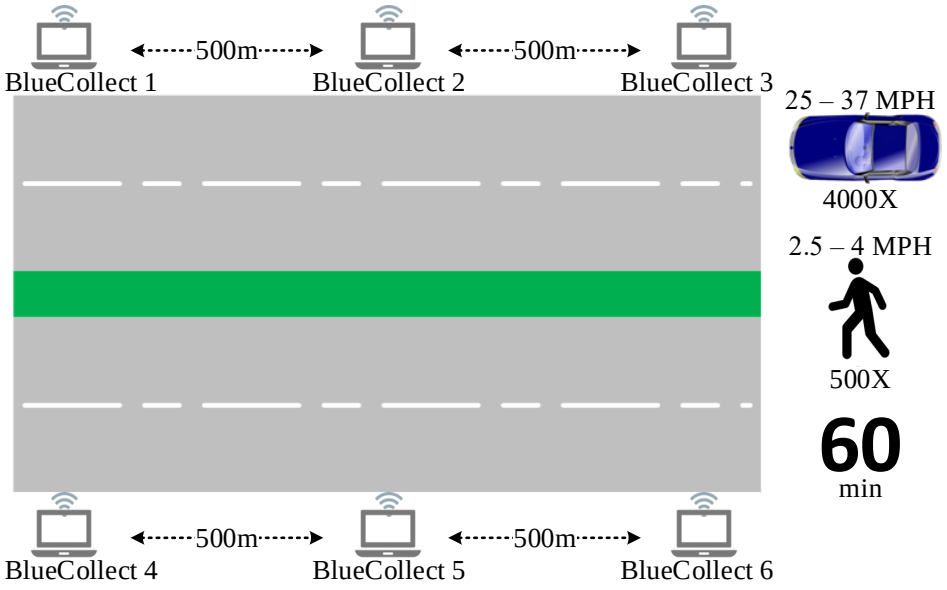


FIG. 37: Pedestrian and Vehicle Differentiation Simulation Scenario No. 1

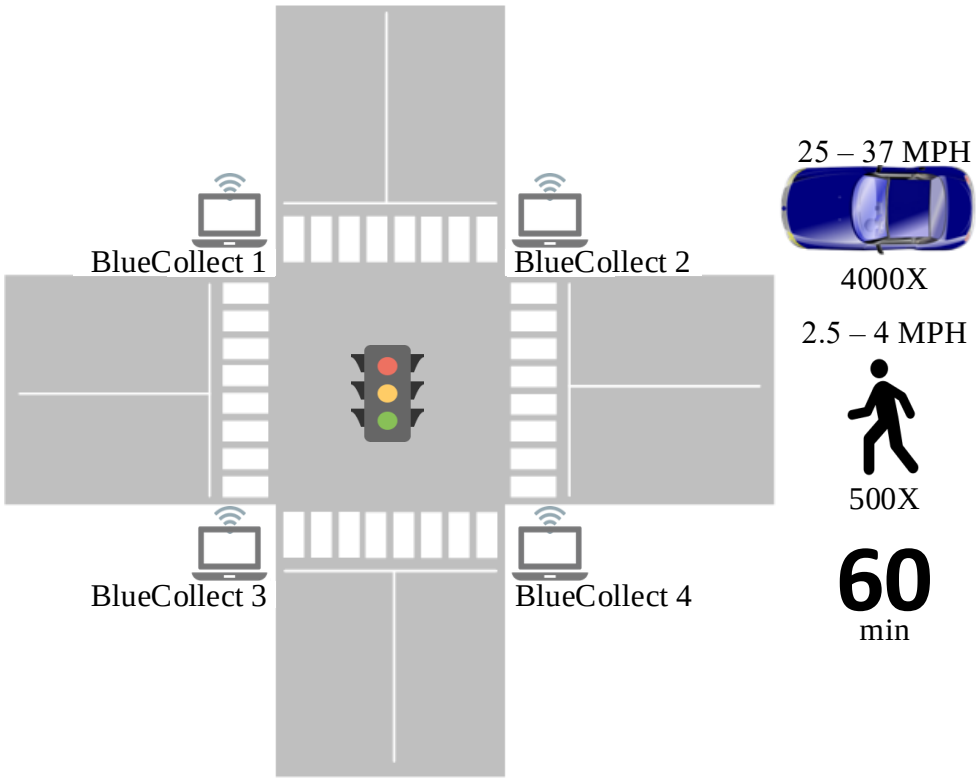


FIG. 38: Pedestrian and Vehicle Differentiation Simulation Scenario No. 2

8.3.4 VEHICLE TRAJECTORIES RECONSTRUCTION EVALUATION

In this subsection, we show the correctness and the completeness of reconstructed vehicle trajectories. As seen in Section 8.2, the vehicle trajectory reconstruction application consumes the Vehicle Detection Locations and Directions on Target Area provided by the itsBlue framework to reconstruct vehicle trajectories. The vehicle location and direction are extracted, relying on number of Bluetooth features. Thus, in this subsection, we show the validity of Bluetooth features that are used to determine vehicle location and direction.

A. Vehicle Location and Direction Determination Features Validation

As seen in Chapter 6, vehicle street segments and directions are determined by exploiting the vehicle Bluetooth features of:

1. Number of received Bluetooth discovery response messages
2. Bluetooth RSS samples mean
3. Bluetooth RSS samples variance

These features are Bluetooth radio signal temporal and spatial characteristics, which means that obtained readings might vary, depending on the street segment length. Therefore, we designed simulation scenarios of three street segment lengths, $120m$, $240m$ and $480m$. In each simulation scenario, there are 500 vehicles traveling at speeds ranging from 30MPH to 45MPH, detected by 25 mobile BlueCollect units carried on buses moving at speeds ranging from 30MPH to 40MPH.

As seen in Figures 39, 40 and 41, the simulation outcomes show that each class features cluster is clearly separated from other class clusters. The number of received Bluetooth messages feature readings separate the three classes, in most cases. Rarely, over short distances, partial overlapping between the readings from vehicles traveling in the opposite direction of a BlueCollect unit carrier and vehicles on other streets may occur, due to a resemblance in the time elapsed while these vehicles are in coverage zone. Thus, we exploit Bluetooth's RSS variance feature, which delivers readings that clearly separate vehicles traveling in opposite direction of the BlueCollect unit carrier from vehicles on nearby streets. In addition, RSS variance readings of vehicles traveling with a BlueCollect carrier on the same street segment and direction may partly overlap with readings of vehicles on other street segments, over

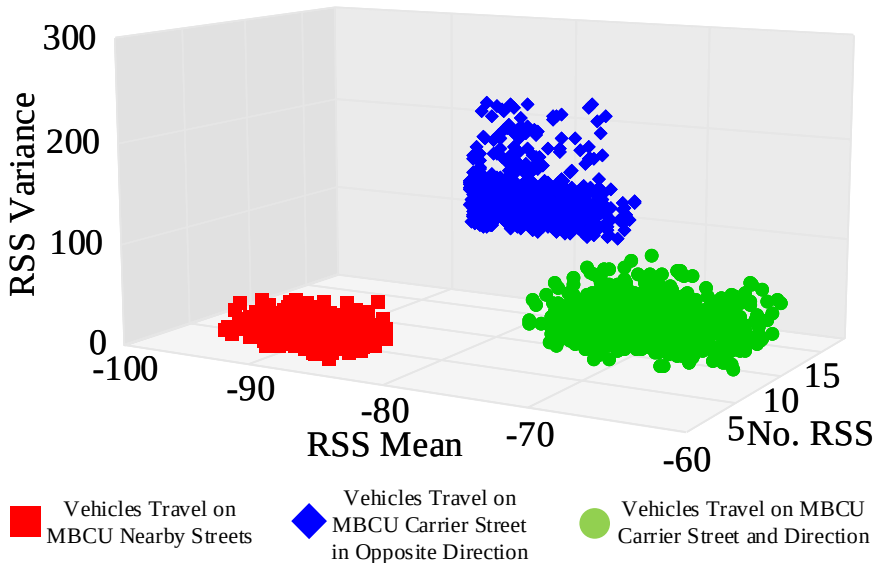


FIG. 39: Bluetooth Features Obtained From Data Collected on a 120m Roadway

short distances. RSS mean readings of vehicles traveling on streets around the mobile BlueCollect unit carrier are noticeably weaker than RSS mean readings of vehicles on the same street segment of the BlueCollect unit carrier, regardless of the direction, which allows them to overcome the minor overlaps caused by RSS variance readings. Additionally, the slight overlap between the RSS mean readings of vehicles moving in the BlueCollect unit carrier direction and vehicles moving in opposite direction over short distances is tackled by the number of Bluetooth message feature readings, which show a high contrast between the readings of these two classes.

B. Vehicle Trajectories Reconstruction Application Evaluation

The vehicle trajectory reconstruction application performance is evaluated using the following criteria:

1. **The correctness of reconstructed vehicle trajectory**. This is expressed by the number of correctly detected street segments over the number of all detected trajectory street segments.

2. **The completeness of reconstructed vehicle trajectory**. This is measured by dividing the number of correctly detected street segments by the actual number of vehicle trajectory street segments.

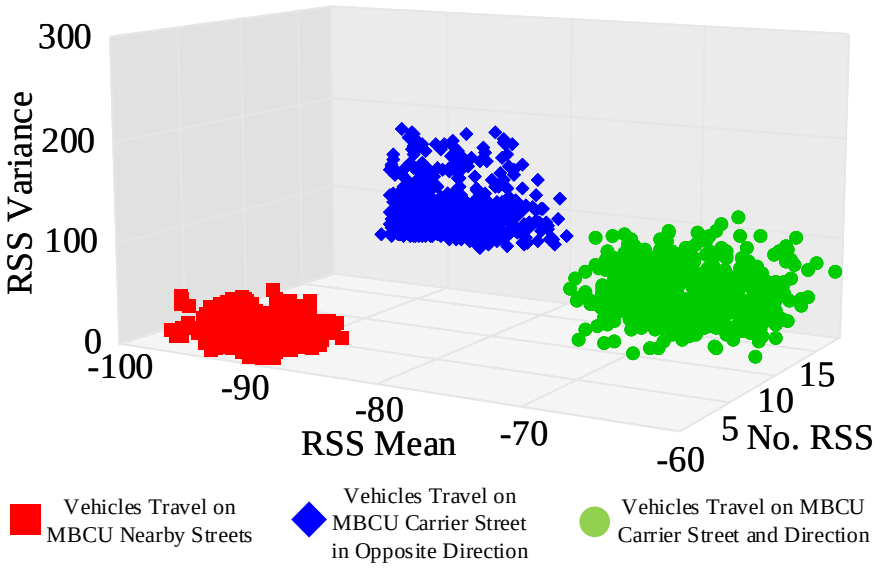


FIG. 40: Bluetooth Features Obtained From Data Collected on a 240m Roadway

Moreover, the evaluation includes an assessment of the impact of the number and carrier type of mobile BlueCollect units on the reconstructed vehicle trajectory correctness and completeness. Also, we highlight the percentage of detected vehicles.

Simulation Scenario

An evaluation dataset is extracted from a PTV VISSIM real-world simulation scenario of downtown Boise, Idaho (FIG. 42). The simulation network contains about 130 street segments connected by more than 50 intersections over $4km^2$. The simulation is run for 15 minutes.

The vehicle trajectory reconstruction application performance's sensitivity to the number and carrier type of BlueCollect units is evaluated by repeating the simulation five times with different numbers and carrier types of BlueCollect units. In the first run, we provide 12 intersections with clusters of eight stationary BlueCollect units for each. The second run is conducted with 12 mobile BlueCollect units carried on buses. In the third run, another 12 mobile BlueCollect units carried on buses are added. After that, for the fourth run, we add 24 mobile BlueCollect units carried on police patrols. Finally, the 12 controlled intersections and the 48 mobile BlueCollect units are used together on the fifth run.

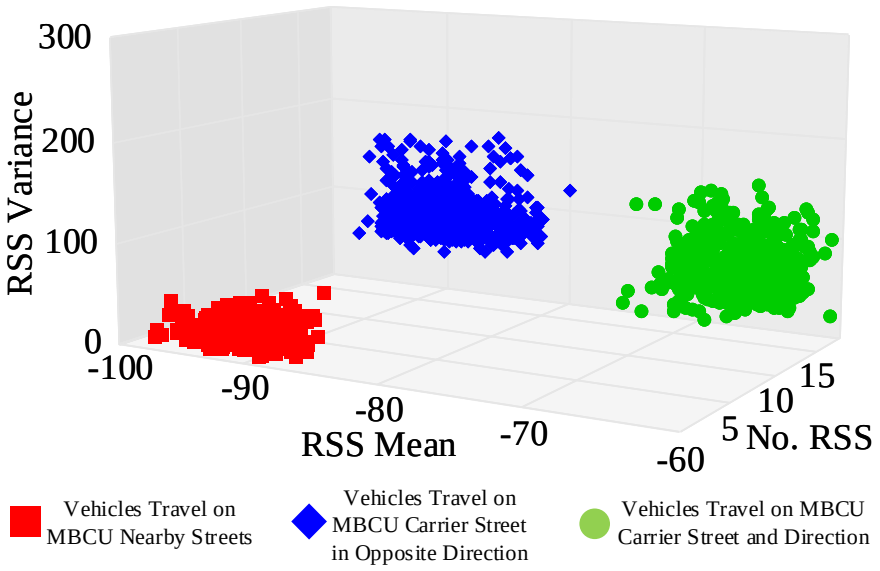


FIG. 41: Bluetooth Features Obtained From Data Collected on a 480m Roadway

Results

The simulator outcomes show a high correctness of reconstructed vehicle trajectories most of the time, due to the precise Bluetooth features used to determine vehicle street segments and direction. The evaluation results show that the correctness of reconstructed vehicle trajectories is above 90% when only 24 mobile BlueCollect units are used (FIG. 43). That number jumps to about 95% when 48 mobile BlueCollect units are in use.

In addition, the vehicle trajectory reconstruction application is able to detect about 86% of the traffic using 48 mobile BlueCollect units (Table 5). Adding 12 intersections with stationary BlueCollect units boosts the vehicle detection percentage to around 91%. In addition, the evaluation results show the advantage of mobile BlueCollect units over the stationary ones in vehicle trajectory reconstruction. 24 mobile BlueCollect units allow the application to reconstruct about 87% of vehicle trajectory (FIG. 44). Adding 24 mobile BlueCollect units carried on police patrols increases the percentage of reconstructed vehicle trajectory to 89%. The data collected by stationery BlueCollect units and provided by the vehicle locations at signalized intersections service boosts that percentage to around 93%.

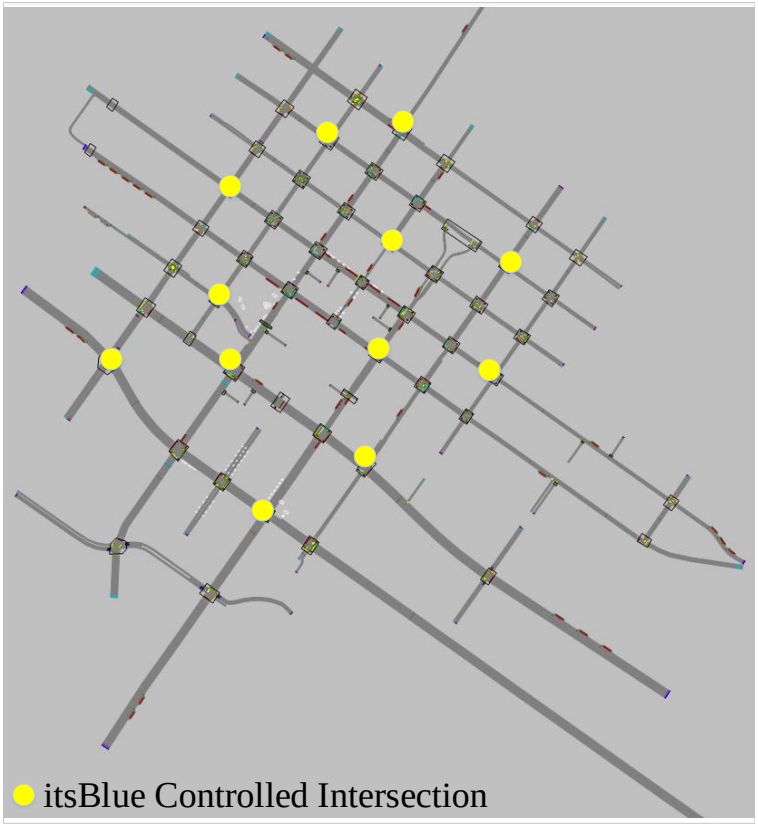


FIG. 42: Simulation Ground Map

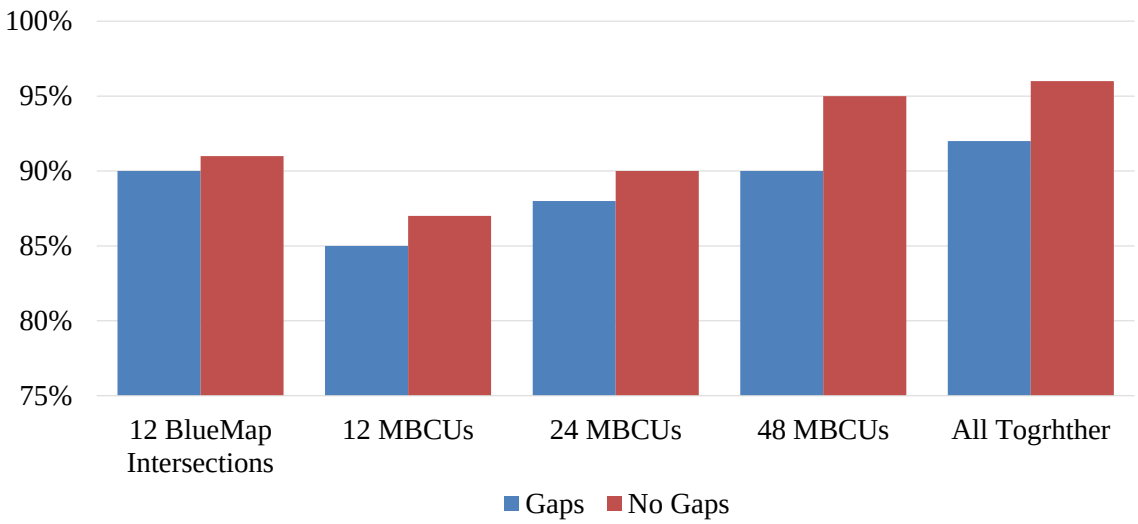


FIG. 43: Reconstructed Vehicle Trajectories Correctness

TABLE 5: Vehicles Detection Percentages with Multiple BlueCollect Units Configurations

12 Intersections	12 MBCUs	24 MBCUs	48 MBCUs	All
54%	42%	67%	86%	91%

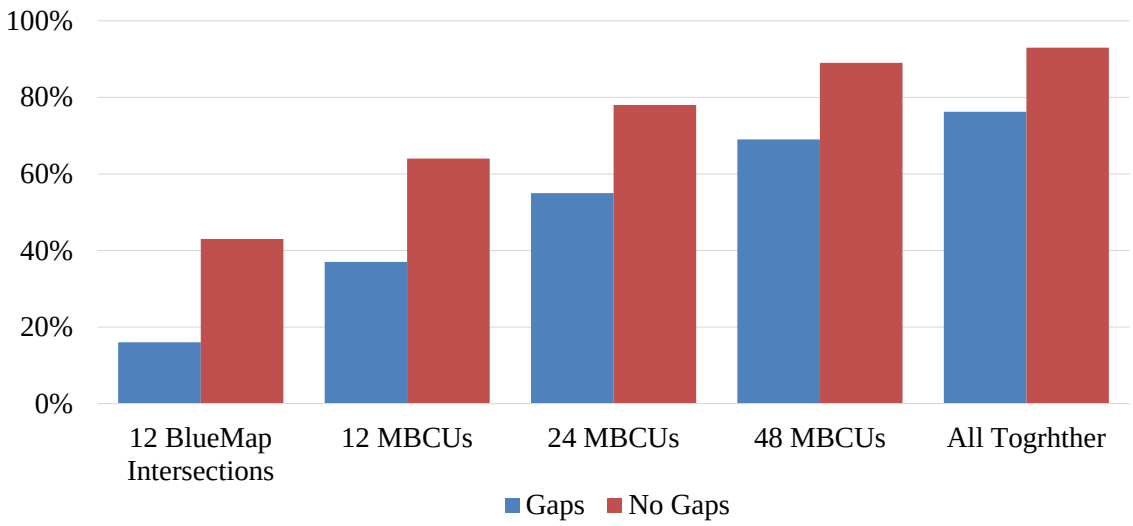


FIG. 44: Reconstructed Vehicle Trajectories Completeness

CHAPTER 9

CONCLUSION AND FUTURE DIRECTIONS

In this chapter, we provide the dissertation conclusion. The conclusion includes a summary of the work that we accomplished and the contribution we added. Then, we present a road map for enhancing and extending our work in the future.

9.1 CONCLUSION

In this dissertation, we presented itsBlue, a novel Bluetooth-based framework to provide ITS researchers and developers with real-time and historical traffic information in an efficient and a cost-effective manner. In itsBlue, we exploit the ubiquity of Bluetooth-enabled devices, cost-effectiveness, Bluetooth data richness and collection easiness, and privacy preservation, to address several challenges that state of the art ITS technologies are facing.

The itsBlue framework collects Bluetooth road users' data and associates it with a data collection location. Then, it utilizes the collected data to extract a variety of road user features such as road user context, appearance time, and vehicle locations and directions. itsBlue allows ITS applications to obtain available traffic information via a set of APIs to facilitate connection and delivery. These operations are carried out in five layers; every layer involves a number of hardware and software components to perform layer tasks.

The first itsBlue layer is the data collection. In itsBlue, we designed and built a compact computing unit called BlueCollect to collect required data. BlueCollect, a computing unit based on a credit card-sized computer, is provided with Bluetooth transceivers, GPS, and wireless communication adapters. The BlueCollect unit works in two modes: stationary, where it is placed on top of traffic lights or light poles, and mobile, where it is carried on buses or on police patrols. BlueCollect units collect Bluetooth data from road users, and employ a radio signal filtering technique to remove signal outliers. In the mobile operation mode, BlueCollect logs GPS coordinates and extracts map location and the direction of data collection. Then, road users' Bluetooth data and collection locations are transferred to the central

computing unit upon recognition of certain spatial or temporal triggers. BlueEngine, the central computing unit, aggregates road users' data and stores it in the database for further manipulation.

In the basic Bluetooth feature layer, the BlueEngine utilizes road user Bluetooth data to extract a number of spatial and temporal features such as the Bluetooth RSS mean, Bluetooth RSS variance, road user appearance time, and number of received Bluetooth discovery response messages.

The third layer is the advanced road user features layer. In this layer, BlueEngine utilizes road user data and basic features to extract traffic-related features. The first extracted feature is the road user context, in which we utilized the divergence between the number of road users' spatial and temporal features to classify them to vehicle riders and pedestrians. To do so, we employed an SVM machine learning technique to classify road users. The second extracted feature is the vehicle locations at signalized intersection. To identify vehicle location at traffic light intersections, we provide BlueEngine with Bluetooth RSS fingerprints of the target intersection to obtain the intersection's RSS distribution. Then, the vehicle location is identified by matching the vehicle's RSS samples with the intersection RSS fingerprints by applying the Bayesian Theorem. To enhance vehicle identification accuracy, the itsBlue framework exploits the vehicle distribution at the intersection. The vehicle spots with a higher likelihood of occupancy are granted extra weight over others in vehicle spot identification. Additionally, to avoid lane blocking and a lack of traffic fluidity, we collected Bluetooth RSS fingerprints of target intersections from moving vehicles in a novel way. The third advanced road user feature extracted in this layer is the moving vehicles' street location and direction. This feature is extracted by using Bluetooth data collected by a Mobile BlueCollect unit. The BlueEngine utilized the location data of Mobile BlueCollect units and vehicles' spatial and temporal features to find vehicle location and direction, according to the Mobile BlueCollect unit that detected it. Vehicles detected by a Mobile BlueCollect units are divided into three groups: vehicles moving with the BlueCollect unit carrier on the same street and direction, vehicles moving with it on the same street in the opposite direction, and vehicles traveling on nearby streets.

Next, we discussed the traffic information provision layer in which itsBlue provides the ITS application with the required traffic information. The itsBlue framework provides the ITS community with the following:

1. Contexts of road users on certain locations and times
2. Vehicle locations at signalized intersection at certain times
3. Vehicle street locations and directions at certain locations and times
4. Location information of Mobile BlueCollect units at certain locations and times
5. Raw Bluetooth data collected from certain location at certain time

To facilitate providing the ITS client with required information, itsBlue implemented a set of APIs using Java RMI, which both allows the ITS client to look up and invoke appropriate APIs to obtain required information, and keeps networking complexity behind the scene.

The top layer is the application layer, where the ITS applications are working. On this layer, ITS applications implement Java RMI clients to invoke itsBlue desired APIs. On this layer, we implemented an intersection management applications pack and a vehicle trajectory reconstruction application.

The intersection management applications pack includes an application to extract vehicle queue lengths at signalized intersections, a vehicle waiting time estimation application, and a pedestrian volume and waiting times determination. Vehicle queue lengths and waiting are extracted from vehicle locations at signalized intersections that provided by itsBlue. These applications analyses obtained information to extract the number of vehicles stopping on every lane at intersections and determine the vehicles' waiting times. The pedestrian volume is extracted from road users' contexts at a target location provided by itsBlue. Pedestrian waiting time to cross is determined by calculating the time elapsed while the pedestrian's Bluetooth device is in coverage of a special purpose short range Bluetooth transceiver placed on the crossing zone.

The vehicle trajectory reconstruction application utilizes vehicle street location and direction provided by itsBlue. In this application, vehicle trajectory is reconstructed using a series of street locations and directions in the target area. This application development involves addressing several challenges in novel ways. The first addressed challenge is the trajectory inconsistency caused by inaccurate vehicle street location or direction. Vehicle trajectory inconsistency is resolved by analyzing vehicle movement and by correcting any inconsistent vehicle trajectory parts, accordingly. Identifying and filling any reconstructed vehicle trajectory gaps is another

challenge that we addressed. To achieve that, we adopted a BFS graph traversal algorithm to develop a trajectory gap identification and filling application. The BFS algorithm is used to find all of the paths that connect gap-bordering intersections. Then, the path with most similar travel time to the time elapsed on the gap is selected to fill the gap. To enhance the performance, the gap filling algorithm excludes any branch with travel time longer than vehicle elapsed time in the gap.

The itsBlue framework and applications are evaluated in various field experiments and simulations. The lack of a reliable Bluetooth simulation tool encouraged us to implement an enhanced version of UCBT NS-2 in which we developed a physical layer to be able to obtain a received Bluetooth RSS. Our simulation package includes a PTV VISSIM to generate large-scale transportation network traces. Then, PTV VISSM traces are converted to UCBT NS-2 to simulate Bluetooth communications.

We summarize our findings from evaluation in the following:

- The RF localization technology employed to identify vehicle locations at signalized intersection showed high performance. The TPR of vehicle locations identified using three RSS samples received by three Bluetooth transceivers reached 96%. The high vehicle location identification accuracy is reflected on beneficiary ITS applications. The vehicle queue length obtained by relying on occupied vehicle locations showed high performance, as well. The TPR of vehicle queue length determined by four Bluetooth transceivers exceeded 90% in about $30s$.

- The utilization of Bluetooth radio signals' spatial and temporal characteristics to extract traffic information is promising. In the following, we highlight our prominent findings:

  - Utilizing the road user Bluetooth features of appearance time, number of received Bluetooth discovery response messages, and RSS variance allowed us to accurately reveal road user context. The vehicle and pedestrian differentiation showed classification TPR of 95% and above in $25s$, for straight road, and needed an additional $20s$ to reach a similar TPR at an intersection.

  - The on-board Bluetooth device features of appearance time, RSS mean, and RSS variance are utilized to discover vehicle location and direction according to the Mobile BlueCollect unit. These Bluetooth features showed

the ability to accurately determine vehicle street location and direction. The number of Bluetooth discovery response messages received from vehicles traveling with the Mobile BlueCollect unit carrier on the same street segment and direction is obviously higher than the number of Bluetooth messages received from vehicles traveling in the opposite direction or on other streets. The variance of Bluetooth RSS samples received from a vehicle traveling in a Mobile BlueCollect carrier in the opposite direction is noticeably higher than those for vehicles traveling on mobile BlueCollect unit carrier in the same direction or on a nearby street. The mean of Bluetooth RSS samples received from vehicle traveling with mobile BlueCollect unit on the same street segment and in the same direction is clearly higher than the RSS mean of Bluetooth samples received from vehicles traveling on other neighboring streets.

- The validation of the RSSI samples obtained by the enhanced version of UCBT NS-2 showed high similarity to the RSS samples obtained from the field experiment

9.2 FUTURE DIRECTIONS

In this section, we discuss future research directions to extend and enhance the work presented in this dissertation.

9.2.1 BLUECOLLECT UNITS INTERCOMMUNICATION AND COORDINATION

The BlueCollect units are independent computing units that are controlled by BlueEngine. BlueCollect units' coordination, such as time synchronization, is handled by itsBlue, which places a high load on BlueEngine. Therefore, we intend to provide the BlueCollect units with intercommunication, in the future. From a different aspect, BlueCollect unit intercommunication opens up the door widely for BlueCollect units' cooperation to enhance data collection.

BlueCollect units' intercommunication will allow us to considerably enhance data collection. One of the features that will be added to enhance the data collection process is the cooperative Bluetooth radio channels distribution. In fact, the Bluetooth

transceiver broadcasts and receives Bluetooth packets on a set of radio channels assigned based on a clock or previously assigned, which may lead to interference and packet loss. To avoid that, we will design an approach that will allow Bluetooth transceivers that are working on certain sites to alternate between radio channels in a synchronized way to avoid interference and packet loss.

Furthermore, BlueCollect units' intercommunication and synchronization will allow traffic information extraction on the BlueCollect level. One of the future extensions that will be added by utilizing BlueCollect units' intercommunication and synchronization is the vehicle street location and direction determination at the BlueCollect level, in which a BlueCollect unit will disseminate a detected vehicle's Bluetooth address and detection time to BlueCollect units on the same street. Once that vehicle has been detected by another BlueCollect unit on the street, the BlueCollect unit will be able to determine the vehicle's street location and direction, relying on vehicle traveled path.

9.2.2 BLUECOLLECT LIGHT VERSION

The data collection unit, BlueCollect, is one of the novelty aspects of itsBlue. BlueCollect's ability to be carried on a vehicle allows for a reduction in the framework's initiation cost and extends the target area. As a step forward, we are planning to develop a smartphone-based BlueCollect. Actually, the wide spread of smartphones worldwide and their capability to collect required data are appealing in their ability to be used to enhance itsBlue data collection. Today's smartphones are equipped with various wireless communication technologies and GPS which, in turn, allows for the collection of itsBlue required data using an intentionally designed app. Despite the drawbacks of participatory sensing, the BlueCollect Light Version is expected to enhance the data collection as a secondary data source, wherein the formerly presented BlueCollect unit is the main data collection component.

BIBLIOGRAPHY

[1] U.S. Department of Transportation, Bureau of Transportation Statistics, "Transportation statistics annual report 2016." Online; https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/TSAR_2016r.pdf, 2016. Accessed: Jul 17, 2017.

[2] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 urban mobility scorecard," 2015.

[3] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011.

[4] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[5] Statista The Statistics Portal, "Number of mobile phone users worldwide from 2013 to 2019." Online; https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/. Accessed: Jul 7, 2017.

[6] California Department of Motor Vehicles, "Autonomous vehicle disengagement reports 2016." Online; https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report_2016. Accessed: Jul 27, 2017.

[7] Alphabet Inc., "Waymo." Online; https://waymo.com/. Accessed: Jul 7, 2017.

[8] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 775–784, Ieee, 2000.

[9] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 205–218, ACM, 2005.

[10] Pew Research Center, "Mobile fact sheet." Online; http://www.pewinternet.org/fact-sheet/mobile/. Accessed: Jul 17, 2017.

[11] Statista The Statistics Portal, "Installed base of bluetooth-enabled devices worldwide in 2012 and 2018." Online; https://www.statista.com/statistics/283638/installed-base-forecast-bluetooth-enabled-devices-2012-2018/. Accessed: Jul 7, 2017.

[12] Business Wire, "Bluetooth technology to be standard in more than 90 percent of automobiles by 2016, strategy analytics study shows bluetooth enabled devices reduce visual distractions in the car." Online; http://www.businesswire.com/news/home/20101118006520/en/Bluetooth-Technology-Standard-90-Percent-Automobiles-2016. Accessed: Jul 17, 2017.

[13] A. Franssens, "Impact of multiple inquires on the bluetooth discovery process: and its application to localization," 2010.

[14] "Bluetooth low energy wireless technology backgrounder." Online; https://www.nordicsemi.com/eng/content/download/3045/40678/version/1/file/Bluetooth_le_technology_backgrounder_4.pdf, 2011. Accessed: July 3, 2015.

[15] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pp. 46–51, IEEE, 2007.

[16] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pp. 232–237, IEEE, 2012.

[17] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, "Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario," in *Wireless Symposium (IWS), 2013 IEEE International*, pp. 1–4, IEEE, 2013.

[18] "SENA Parani UD-100." Online; http://www.senaindustrial.com/products/industrial_bluetooth/ud100.php. Accessed: June 22, 2015.

[19] S. Sunkari, R. Parker, H. Charara, T. Palekar, and D. Middleton, "Evaluation of cost-effective technologies for advance detection," tech. rep., 2005.

[20] J. H. Kell, I. J. Fullerton, and M. K. Mills, "Traffic detector handbook," tech. rep., 1990.

[21] E. Parkany and C. Xie, "A complete review of incident detection algorithms & their deployment: what works and what doesn't," tech. rep., 2005.

[22] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Computer Vision–ECCV 2012*, pp. 645–659, Springer, 2012.

[23] H. Cho, P. E. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 1035–1042, IEEE, 2012.

[24] X. Chang and L. Gao, "Car detection using markov random fields on geometric features," in *Decision and Control, 2007 46th IEEE Conference on*, pp. 2856–2861, IEEE, 2007.

[25] Y. Liu, X. Lu, and J. Xu, "Traffic scenes invariant vehicle detection," in *Control Conference (ASCC), 2013 9th Asian*, pp. 1–6, IEEE, 2013.

[26] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 323–336, ACM, 2008.

[27] I. Ustin, A. Alasaadi, M. Cetin, and T. Nadeem, "Detecting vehicle stops from smartphone accelerometer data," in *The 21st World Congress on Intellegent Transportation Systems, ITSWC2014*, ITSWC, 2014.

[28] Y. Maekawa, A. Uchiyama, H. Yamaguchi, and T. Higashino, "Car-level congestion and position estimation for railway trips using mobile phones," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 939–950, ACM, 2014.

[29] Q. Li, Q. Han, X. Cheng, and L. Sun, "Queuesense: Collaborative recognition of queuing on mobile phones," in *Sensing, Communication, and Networking*

(SECON), 2014 Eleventh Annual IEEE International Conference on, pp. 230–238, IEEE, 2014.

[30] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *Pervasive computing and communications (PerCom), 2013 IEEE international conference on*, pp. 193–200, IEEE, 2013.

[31] M. Cetin, "Estimating queue dynamics at signalized intersections from probe vehicle data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2315, no. 1, pp. 164–172, 2012.

[32] P. Hao and X. J. Ban, "Estimating vehicle position in a queue at signalized intersections using sample travel times from mobile sensors."

[33] J.-Q. Li, K. Zhou, S. E. Shladover, and A. Skabardonis, "Estimating queue distance under connected vehicle technology: Using probe vehicle, loop detector, and fused data," *Journal of the Transportation Research Board*, pp. 17–22, 2013.

[34] S. M. Turner, W. L. Eisele, R. J. Benz, and D. J. Holdener, "Travel time data collection handbook," tech. rep., 1998.

[35] A. Pathak, A. Jindal, Y. C. Hu, and S. P. Midkiff, "What is keeping my phone awake?: characterizing and detecting no-sleep energy bugs in smartphone apps," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 267–280, ACM, 2012.

[36] Y. Malinovskiy, Y.-J. Wu, Y. Wang, and U. K. Lee, "Field experiments on bluetooth-based travel time data collection," in *Transportation Research Board 89th Annual Meeting*, no. 10-3134, 2010.

[37] B. N. Araghi, L. Tørholm Christensen, R. Krishnan, J. Hammershøj Olesen, and H. Lahrmann, "Improving accuracy of bluetooth-based travel time estimation using low-level sensor data," in *Transportation Research Board 92nd Annual Meeting, number*.

[38] M. Asudegi and A. Haghani, "Optimal number and location of node-based sensors for collection of travel time data in networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2338, no. 1, pp. 35–43, 2013.

[39] P. Songchitruksa and D. Middleton, "Enhancing work zone travel time estimation using smart work zone data," in *Transportation Research Board 92nd Annual Meeting*, no. 13-0989, 2013.

[40] A. M. Hainen, J. S. Wasson, S. M. Hubbard, S. M. Remias, G. D. Farnsworth, and D. M. Bullock, "Estimating route choice and travel time reliability with field observations of bluetooth probe vehicles," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2256, no. 1, pp. 43–50, 2011.

[41] S. Bohonos, A. Lee, A. Malik, C. Thai, and R. Manduchi, "Universal real-time navigational assistance (urna): an urban bluetooth beacon for the blind," in *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, pp. 83–88, ACM, 2007.

[42] NovAtel, "Enclosures flexpak6." Online; https://www.novatel.com/assets/Documents/Papers/FlexPak6.pdf. Accessed: Jul 7, 2017.

[43] T. Walter, "Satellite based augmentation systems," in *Springer Handbook of Global Navigation Satellite Systems*, pp. 339–361, Springer, 2017.

[44] "Raspberry pi 2 model b." Online; https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. Accessed: September 10, 2015.

[45] "Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. - part 15.1: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–580, 2005.

[46] Bluetooth, SIG, "Specification of the bluetooth system, version 4.2," *Dec*, 2014.

[47] J. Liu, C. Chen, and Y. Ma, "Modeling and performance analysis of device discovery in bluetooth low energy networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 1538–1543, IEEE, 2012.

[48] N. CINEFRA, "An adaptive indoor positioning system based on bluetooth low energy rssi," 2014.

[49] "Google maps developer's guide." Online; https://developers.google.com/maps/documentation/geocoding/intro. Accessed: Mar 27, 2017.

[50] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[51] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.

[52] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, pp. 1137–1145, 1995.

[53] L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, and Y. Chen, "Using inquiry-based bluetooth rssi probability distributions for indoor positioning," *Journal of Global Positioning Systems*, vol. 9, no. 2, pp. 122–130, 2010.

[54] "Remote method invocation (rmi)." Online; http://www.oracle.com/technetwork/articles/javaee/index-jsp-136424.html. Accessed: May 2, 2015.

[55] Q. Wang and D. Agrawal, "Ucbt-bluetooth extension for ns2 at the university of cincinnati," 2013.

[56] PTV, AG, "Vissim 5.20 user manual," *Karlsruhe, Germany*, 2009.

[57] K. Fall and K. Varadhan, "The network simulator (ns-2)," *URL: http://www.isi. edu/nsnam/ns*, 2007.

VITA

AHMED AWAD ALGHAMDI
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

Ahmed received his Bachelor of Science in Computer Science with honors in 2005 from the College of Computer and Information Sciences at Al Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. He then worked as a Teaching Assistant for the Department of Computer Science at Al Imam Mohammad Ibn Saud Islamic University from June 2005 to April 2007. In 2009, he received his Masters Degree in Software Systems Engineering from the School of Computing and Information Systems at the University of Melbourne, Australia. During the course of his Ph.D. study, he has co-authored three scientific publications. He is a member of IEEE and ACM.

Typeset using LaTeX.